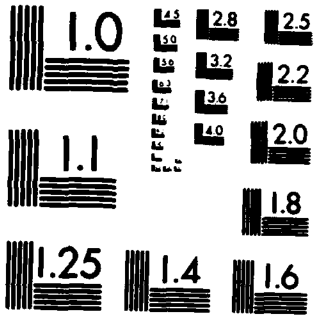
AD-A125 302 COMPUTER SIMULATION OF A MULTIAXIS AIR-TO-AIR TRACKING 1/2
TASK USING THE OPT..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH W A YUCUIS DEC 82
UNCLASSIFIED AFIT/CI/NR-82-62T F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1

AD A125302

DTIC FILE COPY

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR-82-62T	2. GOVT ACCESSION NO. AD-A125302	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Computer Simulation of a Multiaxis Air-to-Air Tracking Task Using the Optimal Pilot Control Model		5. TYPE OF REPORT & PERIOD COVERED THESIS/DISSERTATION
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) William Allen Yucuis		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: Purdue University		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433		12. REPORT DATE Dec 82
		13. NUMBER OF PAGES 110
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASS
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17

17 Feb 83

Lynn Wolaver
LYNN E. WOLAVER
Dean for Research and
Professional Development
AFIT, Wright-Patterson AFB OH

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

ATTACHED

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

83 03 03 038

ABSTRACT

Yucuis, William Allen. M.S., Purdue University, December 1982. Computer Simulation of a Multiaxis Air-to-Air Tracking Task Using the Optimal Pilot Control Model. Major Professor: David K. Schmidt.

The primary objective of the research is to simulate the multiaxis air-to-air tracking task and determine if the augmentation control synthesis results in improved piloted vehicle performance. Also, the simulation results will be compared to previously obtained analytical results to see if the optimal pilot control model should be modified. The simulation was accomplished on the Aeronautical Engineering minicomputer simulation system and the results substatiate the claim of improved performance. Since the analytical model and simulation task have different target motions, nothing definitive can be said about modifying the model.

AFIT/CI/NR 82-62T

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to ascertain the value and/or contribution of research accomplished by students or faculty of the Air Force Institute of Technology (ATC). It would be greatly appreciated if you would complete the following questionnaire and return it to:

AFIT/NR
Wright-Patterson AFB OH 45433

RESEARCH TITLE: Computer Simulation of a Multiaxis Air-to-Air Tracking Task Using them Optimal Pilot Control Model

AUTHOR: William Allen Yucuis

RESEARCH ASSESSMENT QUESTIONS:

1. Did this research contribute to a current Air Force project?

( ) a. YES ( ) b. NO

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not?

( ) a. YES ( ) b. NO

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency achieved/received by virtue of AFIT performing the research. Can you estimate what this research would have cost if it had been accomplished under contract or if it had been done in-house in terms of manpower and/or dollars?

( ) a. MAN-YEARS __________ ( ) b. $__________

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3. above), what is your estimate of its significance?

( ) a. HIGHLY SIGNIFICANT ( ) b. SIGNIFICANT ( ) c. SLIGHTLY SIGNIFICANT ( ) d. OF NO SIGNIFICANCE

5. AFIT welcomes any further comments you may have on the above questions, or any additional details concerning the current application, future potential, or other value of this research. Please use the bottom part of this questionnaire for your statement(s).

NAME GRADE POSITION

ORGANIZATION LOCATION

STATEMENT(s):

A

FOLD DOWN ON OUTSIDE - SEAL WITH TAPE

AFIT/NR
WRIGHT-PATTERSON AFB OH 45433

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE. $300

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 73236 WASHINGTON D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

AFIT/ DAA
Wright-Patterson AFB OH 45433

FOLD IN

S) 62T

COMPUTER SIMULATION OF A MULTIAXIS
AIR-TO-AIR TRACKING TASK
USING THE OPTIMAL PILOT CONTROL MODEL

A Thesis
Submitted to the Faculty
of
Purdue University
by
William Allen Yucuis

In Partial Fulfillment of the
Requirements for the Degree
of
Master of Science in
Aeronautical and Astronautical Engineering

December 1982

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Yucuis, William Allen. M.S., Purdue University, December 1982.
Computer Simulation of a Multiaxis Air-to-Air Tracking Task Using
the Optimal Pilot Control Model. Major Professor: David K. Schmidt.

The primary objective of the research is to simulate the multiaxis air-to-air tracking task and determine if the augmentation control synthesis results in improved piloted vehicle performance. Also, the simulation results will be compared to previously obtained analytical results to see if the optimal pilot control model should be modified. The simulation was accomplished on the Aeronautical Engineering minicomputer simulation system and the results substatiate the claim of improved performance. Since the analytical model and simulation task have different target motions, nothing definitive can be said about modifying the model.

CHAPTER 1. INTRODUCTION

1.1 Motivation

The trend in future fighter-type aircraft design is towards the designing and building of lighter, more maneuverable aircraft. To attain these goals, designers have turned to new advanced in technology, such as active flutter suppression control systems, direct-lift or side-lift control surfaces, and lighter, yet stronger structural materials. The use of such advanced methods, however, radically alters the performance characteristics of the piloted vehicle, and classical methods for predicting these characteristics have proved inadequate. The classical methods use some type of specification on the vehicle rigid body mode that has traditionally resulted in acceptable piloted vehicle performance. The advanced methods mentioned above, however, introduce higher order dynamics into the system, which are assumed insignificant in classical approaches, and are a cause for errors. Therefore, some other method, or combination of methods, must be used to accurately predict the piloted vehicle performance when higher order dynamics are present. One such approach, which will be tested in this research, uses the optimal pilot control model (OCM),[5] to directly predict piloted vehicle performance. The main purpose of this research is to simulate a multiaxis air-to-air tracking task, which includes higher order dynamics, and to compare the simulation results to already available analytical predictions. The aim is to validate the control synthesis methodology used to obtain an augmentated control system and to determine if any changes to the OCM are required. The analytical development of the control synthesis has been done by Schmidt,(1)(2) and provides the theoretical foundation for the research.

1.2 Outline

The report will begin by presenting an analytical development of the optimal control pilot model and then explaining the control law synthesis used to develop an augmentation system. Chapter 2 will review the basic OCM and then apply the control synthesis to a pitch tracking and a multiaxis tracking task. The analytical study involves a fighter type aircraft with two different sight configurations, which introduce higher order dynamics into the system. Finally, some additional target motion is added to the basic model to more realistically simulate the task.

Chapter 3 outlines the simulation system used to accomplish the simulation. The various hardware components and their capabilities will first be present and then the development of the computer software will be discussed.

Chapter 4 explains how the multiaxis tracking task simulation is used to test the research hypotheses. The assumptions used in the simulation are examined and the evaluation process is explained. An outline for operating the simulation system is given, along with the methodology for analyzing the data. Finally, a summary of simulation and analytical results is tabulated and discussed.

The final chapter, Chapter 5, uses the results given in Chapter 4 to derive some conclusions. The results validate the primary hypothesis of improved piloted vehicle performance using the augmentation control synthesis. The results do not yield any definitive conclusions about changing any of the OCM parameters, since the addition of the target motion makes an accurate comparison impossible. However, the simulation results do give some promising trends, indicating the OCM is a good model of the task. Finally, some suggestions for future research are given.

CHAPTER 2. MODEL DEVELOPMENT

2.1 Introduction

The optimal control pilot model (OCM)[5] has been used successfully to accurately predict piloted vehicle performance.[3] Since it can be readily adapted to a problem that includes higher order dynamics, the OCM was chosen to model the multiaxis air-to-air tracking task. An augmentation control synthesis methodology has been developed by Schmidt[2] which uses the optimal pilot control model as the system model. This chapter will present a brief overview of the OCM and the control synthesis and then apply them to both a pitch tracking and a multiaxis tracking task. A comparison between analytical and simulation results for the unaugmented pitch tracking task, as well as augmented versus unaugmented analytical results for both tasks, will show the advantage of using the OCM and also indicate that the augmented system could have significant improvements in piloted vehicle performance. Finally, some changes to the OCM will be made to more realistically simulate the target vehicle motion.

2.2 Optimal Pilot Control Model and Control Synthesis

The multivariable optimal control pilot model was selected because it has several important advantages. First, the model easily adapts itself to the higher order dynamics present in the problem. Secondly, the OCM optimizes the piloted vehicle performance, which is the ultimate goal of any aircraft designer. The OCM accomplishes this by not only including the vehicle characteristics in the model, but also incorporating the pilot's "strategy" for achieving the desired performance criteria. Finally, the interactive structure between the pilot and the augmentation control system is embedded within the model. This structure accruately reflects both the cooperative nature of a pilot controlling an augmented vehicle and the pilot's ability to readily adapt to various control levels.

The optimal control pilot model represents the system in the familiar linear form of $\dot{\bar{x}} = A\bar{x} + B\bar{u}_p$. The OCM then assumes that a highly trained, well motivated pilot will adopt a "strategy" of choosing the control inputs, $\bar{u}_p$, to minimize an objective cost function, J_p, which is given by

$$J_p = E\{\lim_{T\to\infty} \frac{1}{T} \int_0^T (\bar{y}^T Q\bar{y} + \bar{u}_p^T R\bar{u}_p + \dot{\bar{u}}_p^T G\dot{\bar{u}}_p)dt\} \qquad (2.2.1)$$

The matrices, Q and R, are weightings on the pilot's observation vector, $\bar{y}$, and the control inputs, $\bar{u}_p$, while the matrix G is selected to obtain a specified neuromuscular lag, τ_N.(5) A control law synthesis method is used to improve the piloted vehicle's performance.(1)(2) The augmented system is first represented by $\dot{\bar{x}} = A\bar{x} + B\bar{u}_p + B\bar{u}_a + \bar{w}$, where $\bar{u}_a$ is the equivalent control system input, and $\bar{w}$ is system noise. The augmented control inputs, $\bar{u}_a$, are then selected to minimize an augmented objective cost function, J_a, which is given by

$$J_a = J_p + E\{\lim_{T} \frac{1}{T} (\bar{u}_a^T F\bar{u}_a)dt\} \qquad (2.2.2)$$

The values of $\bar{u}_a$ that minimize J_a are

$$\bar{u}_a = -K_x\,\bar{x} - K_u\,\bar{u}_p \tag{2.2.3}$$

where

$$K_x = F^{-1}B^TK_{A1} \tag{2.2.4}$$

$$K_u = F^{-1}B^TK_{A2} \tag{2.2.5}$$

and K_{A1} and K_{A2} are obtained by simultaneously solving two coupled Ricatti equations.(2) The resulting system is therefore represented by

$$\dot{\bar{x}} = A_{aug}\bar{x} + B_{aug}\bar{u}_p \tag{2.2.6}$$

where

$$A_{aug} = A-BK_x \tag{2.2.7}$$

$$B_{aug} = B-BK_u \tag{2.2.8}$$

The next step is to determine the unique objective cost function for the particular task in question and then apply the augmentation synthesis. The first task to be considered is a pitch-only tracking task.

2.3 Pitch Tracking Task

The initial task to be examined is a pitch-only tracking task,(2) where a fighter type aircraft, employing two significantly different tracking sights, attempts to track a wings level target. This task was selected because the inclusion of the sight dynamics introduce higher order dynamics into the system. Therefore, with the same vehicle rigid body dynamics, the introduction of higher order system dynamics will significantly alter the piloted vehicle performance. This section will first explain the differences between the two sight configurations, and how the higher order dynamics are introduced into the problem, and then present the basic and augmented model parameters, with an explanation of how the parameters were selected.

Any aircraft tracking sight system uses the known ballistics of the attacking aircraft's weapons system plus the available data for the attacker and target states, to "predict" where the next bullet would hit if the gun were fired at this moment. Therefore, if the target is located within the sight, and the gun is fired, a bullet would theoretically impact the target. The accuracy of a sight system is therefore dependent on how accurately the ballistics and states are known. A typical assumption is that the ballistics and the attacker's states are correctly known, so that the target states are the only variables. The two sight configurations, then represent different methods of calculating the target's states. The first sight, called an ideal sight, assumes the target's true line of sight rate, $\dot{\beta}$, and its true normal acceleration, a_T, are known. The displayed sight lead angle, λ, is then calculated by

$$\lambda = T_f(\dot{\theta}-\dot{\beta}) - \frac{T_f^2}{2D} a_T - .038 \frac{VT_f}{D} \alpha \qquad (2.3.1)$$

where $\dot{\theta}$, α, and V are the attacker's pitch rate, angle of attack, and velocity, respectively, D is the target range, and T_f is the projectile time of flight.[(1)] The second sight, called the typical sight, approximates $\dot{\beta}$ and a_T by assuming $\dot{\beta}$ equals $\dot{\lambda}$, the displayed lead angle rate, and $a_T \doteq Z_\alpha \alpha$, the attacker's normal acceleration. The pitch tracking task involves minimizing ε, the tracking error, where $\varepsilon = \lambda-\beta$, using a simple K/s^2 plant. The pitch tracking task display is shown in

Figure 1. The fixed reference point represents the weapon's boreline, while the sight is dynamic, and moves in the display. It is these sight dynamics that introduce the higher order systems dynamics into the problem. By comparing the transfer functions of $\epsilon(s)/\delta_{st}(s)$ for the two sight configurations, as given in Table 1, it should be noted that the numerator coefficients are functions of both the vehicle and sight characteristics, while the denominator coefficients, except for T_f in the typical sight configuration, are functions of only the vehicle characteristics. Therefore, with the two sights introducing higher order system dynamics, a unique objective cost function must be found that allows the OCM to represent this system.

The problem of determining a unique objective cost function for a pitch tracking task was done by Harvey.(3) The pitch tracking task was flown using several actuator dynamics and the two sight configurations mentioned above. Harvey then used fixed base simulation data to determine a unique objective cost function. Harvey's simulation data was for an aircraft with the typical sight configuration, and a comparison between analytic and simulation results is presented in Table 2. Note the excellent correlation between the experimental results and the typical sight analytical predictions, plus the improvements resulting from the ideal sight configuration. He achieved excellent correlation over a wide range of dynamics, so his OCM parameters, as shown in Table 3, were used for applying the augmentation strategy. The augmentation synthesis will now be applied to the OCM to analytically predict improvement in piloted vehicle performance.

The augmentation control synthesis outlined in Section 2.2 was applied to both the ideal and typical sight configurations. The

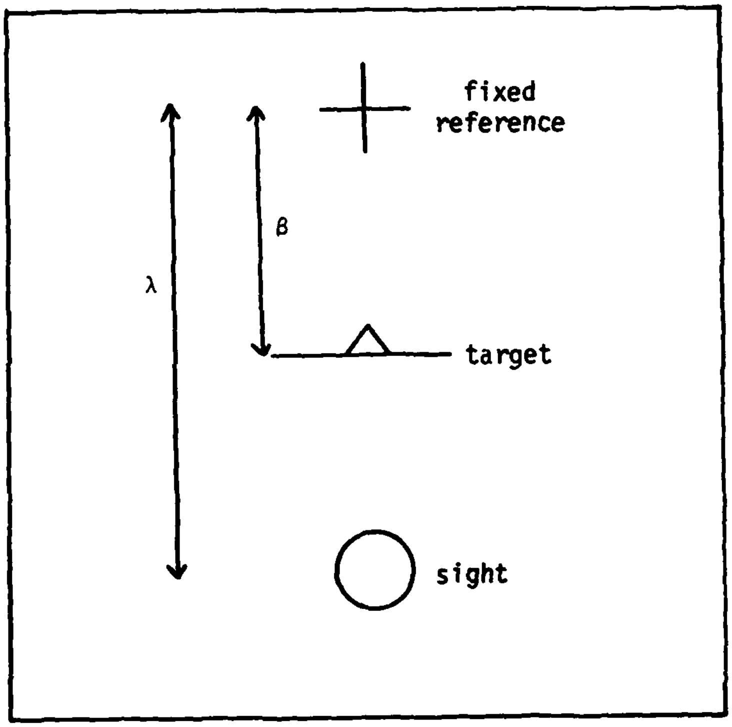


Figure 1

Pitch Tracking Task Display

Table 1

Transfer Function Comparison

Second order plant:

$$\frac{\varepsilon(s)}{\delta_{st}(s)} = \frac{11.7}{s^2}$$

Ideal sight display:

$$\frac{\varepsilon(s)}{\delta_{st}(s)} = \frac{K(N_1 s^3 + N_2 s^2 + N_3 s + N_4)}{s^2(\tau_a s + 1)(s^2 + 2\xi_{sp} w_{sp} s + w_{sp}{}^2}$$

Typical sight display:

$$\frac{\varepsilon(s)}{\delta_{st}(s)} = \frac{K(M_1 s^3 + M_2 s^2 + M_3 s + M_4)}{s^2(T_f s + 1)(\tau_a s + 1)(s^2 + 2\xi_{sp} w_{sp} s + w_{sp}{}^2)}$$

$K = -16\ \tau_a/D$

$N_1 = -.9627\ Z_\delta T_f$

$N_2 = (D/V + T_f)(VM_\delta + Z_\delta M_{\dot{\alpha}}) - Z_\delta$

$N_3 = Z_\delta(M_q + M_{\dot{\alpha}}) + (D/V + T_f)(M_\alpha Z_\delta - Z_\alpha M_\delta)$

$N_4 = M_\alpha Z_\delta - Z_\alpha M_\delta$

$M_1 = Z_\delta T_f\ (Z\alpha D/2V^2 - .962)$

$M_2 = N_2 + T_f(Z\alpha D/2V^2 - .962)(VM_\delta - M_q Z_\delta)$

$M_3 = N_3$

$M_4 = N_4$

Table 2

RMS Performance Comparison

$D = 1000$ ft, $T_T = 5$ g	ϵ(deg)	λ(deg)	q(deg/sec)	δE(deg)
Simulation	2.7	4.0	11.2	3.0
Typical Sight(Analytical)	3.0	4.2	12.1	3.0
Ideal Sight (Analytical)	1.9	3.7	11.0	2.9

$D = 3000$ ft, $T_T = 5$ g	ϵ(deg)	λ(deg)	q(deg/sec)	δE(deg)
Simulation	2.7	12.1	9.2	2.4
Typical Sight (Analytical)	2.7	14.2	10.2	2.5
Ideal Sight (Analytical)	1.4	13.8	8.5	2.3

Table 3

Pilot Model Parameters

Observation delay , $\tau = 0.2$ sec.

Neuromuscular time constant, $\tau_N = 0.1$ sec.

Observation vector, $\bar{y}^T = (\varepsilon, \dot{\varepsilon}, \lambda, \dot{\lambda})$

Cost function weightings, $Q_{y_{ii}} = (16, 1, 0, 4)$

$Ru = 0$

Full attentional allocation

Observation thresholds, $T_\varepsilon = T_\lambda = 0.65$ deg

$T_{\dot{\varepsilon}} = T_{\dot{\lambda}} = 1.3$ deg/sec

Observation noise variance, $V_{y_i} = \pi\rho_y T^2_{y_i}$

$\rho_y = 0.01$

Motor noise variance, $V_{u_p} = \pi\rho_u T^2_{u_p}$

$\rho_u = 0.001$

calculated gains, plus performance results, are tabulated in Tables 4 and 5, and as hoped, the augmented system resulted in a lower error, ϵ, as well as a reduced stick deflection, δ_{st}, and stick rate, $\dot{\delta}_{st}$. Therefore, the control synthesis predicts not only an improvement in performance, as evidenced by reduced error, but also indicates a lowering of pilot workload, as shown by reduced stick deflections and rates. The control synthesis should more accurately predict pilot vehicle performance than the classical methods and this can be shown by comparing root locus plots. Root locus plots for the ideal and typical sight configurations are shown in Figures 2 and 3, respectively, where u shows the unaugmented system roots, and A-C indicate the root locations for increasing levels of augmentation. Notice that the vehicle short period root movement is completely different for the two sight configurations. Since the same vehicle dynamics are present in both cases, the only difference between the two cases would be the higher order dynamics introduced by the different sights. Since classical methods ignore these higher order dynamics, any augmentation strategy based on classical approaches may or may not result in the proper root movement. The piloted vehicle performance characteristics are different for the two sight configurations, so the higher order dynamics must obviously be included in the model in order to accurately predict piloted vehicle performance. The use of the control law synthesis, therefore, appears justified, and a similar analysis will now be done for the more difficult multiaxis tracking task.

Table 4

Augmentation Gains and Performance

IDEAL SIGHT

Optimal Gains

AUGMENTATION LEVEL	K_n	K_{a_T}	K_{r_T}	K_β	K_α	K_Θ	K_q	K_{δ_E}	$K_{\delta_{st}}$
A	$-1.13x10^{-6}$	$-7.35x10^{-6}$	.021	.030	.030	-.051	-.011	.017	.029
B	$-7.25x10^{-6}$	$-4.92x10^{-5}$	.144	.213	.189	-.357	-.072	.111	.154
C	$-2.67x10^{-5}$	$-2.06x10^{-4}$	.632	.995	.734	-1.63	-.314	.432	.383

Performance(rms) - D = 1000 ft., T_T = 5 g.

	Unaugmented	Level A	Level B	Level C
Error, ϵ(deg)	1.92	1.71	0.95	0.35
Deflection, δ(deg)	3.61	3.55	3.50	3.55
Rate, $\dot{\delta}$(deg/sec)	7.16	6.25	5.39	4.87

Table 5

Augmentation Gains and Performance

TYPICAL SIGHT

Optimal Gains

AUGMENTATION LEVEL	K_n	K_{a_T}	K_{r_T}	K_β	K_λ	K_α	K_Θ	K_q	K_{δ_E}	$K_{\delta_{st}}$
A	-1.38×10^{-6}	-9.52×10^{-6}	.027	.037	.014	.042	-.065	-.018	.029	.044
B	-8.19×10^{-6}	-5.90×10^{-5}	.176	.246	.102	.245	-.442	-.122	.185	.211
C	-2.98×10^{-5}	-2.36×10^{-4}	.731	1.07	.493	.872	-1.80	-.535	.689	.467

Performance(rms) - D = 1000 ft., T_T = 5 g.

	Unaugmented	Level A	Level B	Level C
Error, ε(deg)	3.05	2.50	1.12	0.31
Deflection, δ(deg)	3.71	3.64	3.51	3.68
Rate, $\dot{\delta}$(deg/sec)	7.33	6.64	5.36	5.10

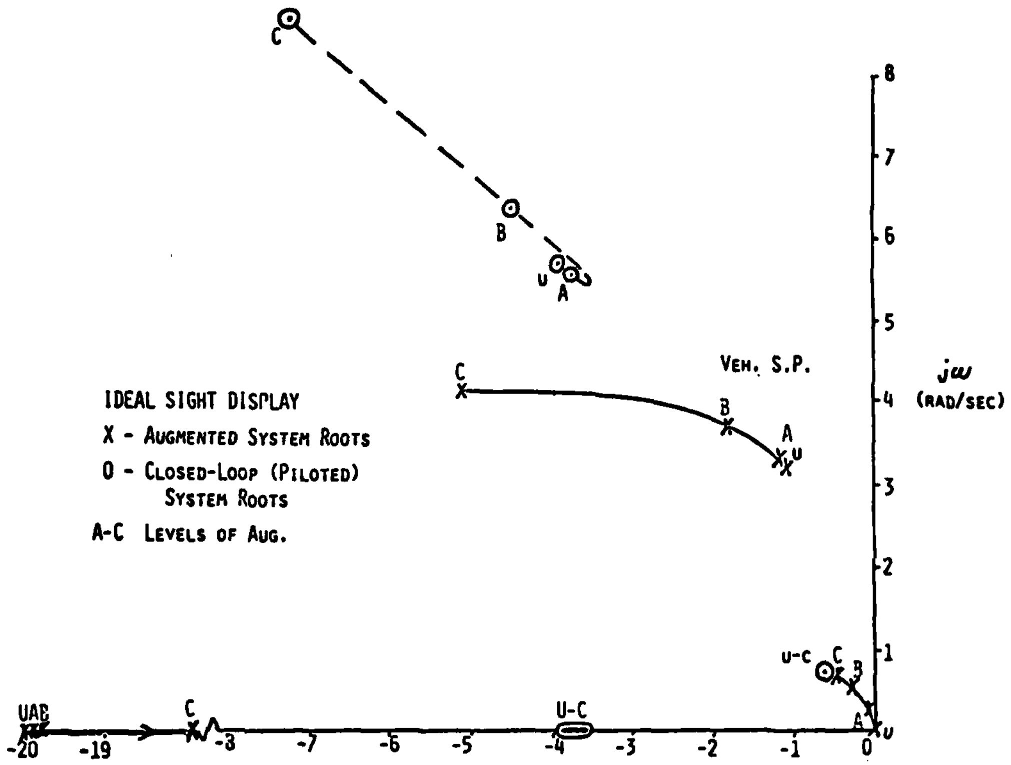


Figure 2

Root Locus (Ideal Sight)

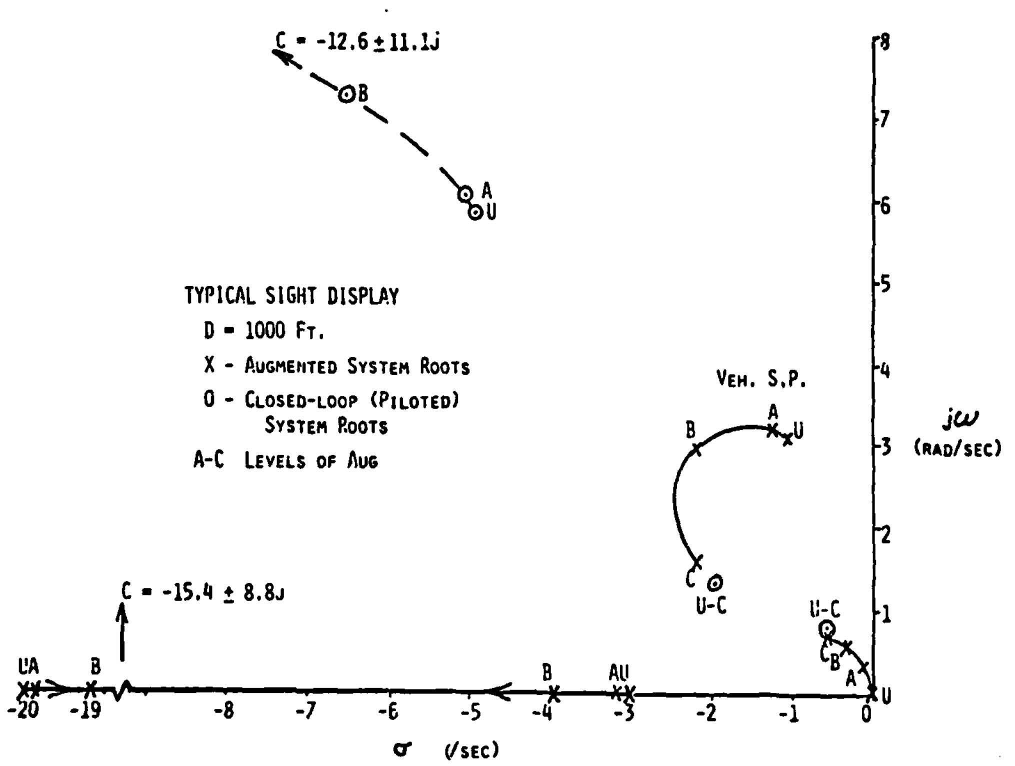


Figure 3

Root Locus (Typical Sight)

2.4 Multiaxis Tracking Task

The multiaxis air-to-air tracking task is very realistic of the type of task required of fighter aircraft. As is normally the case, more realism generally translates into more difficulty, and such is the case here. The difficulty of the multiaxis tracking task arises because of a high degree of unsymmetric coupling between the azimuth and elevation axes. The few previous studies on multiaxis tracking tasks have primarily been concerned with wings-level, position tracking tasks.(6) In that type of task, the dynamics are uncoupled, and each axis can be considered independently. The resulting models are therefore fairly small, and very manageable. This section will discuss how the pitch-only OCM is expanded to account for the interaction between the axes and develop the parameters for the unique objective cost function associated with the multiaxis task. Then, as in the pitch tracking task, the augmentation control synthesis will be applied to determine what improvement, if any, is possible in the piloted vehicle performance.

As before, the system is represented in state variable form as $\dot{\bar{x}} = A\bar{x} + B\bar{u}$ and $\bar{y}(t+\tau) = C\bar{x} + D\bar{u}$. The required matrices are obtained by linearizing the dynamics of an F-106 aircraft, using an ideal sight, about a four-g, turning flight condition, to obtain the vehicle perturbations. The engagement parameters are presented in Table 6, while the aircraft state, control input, and pilot's observation vectors are defined in Table 7. A more complete development of these matrices is contained in Schmidt's report,(1) and the resulting system matrices are shown in Appendix A. A baseline optimal control model was initially obtained as a direct extension of the pitch tracking task model. The

Table 6

Engagement Parameters

Target altitude, h_T = 10,000 ft.

Mach = 0.72

Target velocity, Uss_T = 775 ft/sec

Target flight path angle, Θ_T = 0 deg

Target bank angle, ϕ_T = 75.5 deg

Distance to target, d = 2000 ft.

Target/attacker relative heading, $\Delta\psi$ = 23.6 deg

Target normal acceleration, A_{z_T} = -128.8 ft/sec^2

Target accelerations (attacker's coordinates)

A_{T_x} = -50.0 ft/sec^2

A_{T_y} = - 3.0 ft/sec^2

A_{T_z} = -119.0 ft/sec^2

Target velocities (attacker's coordinates)

V_{T_x} = 710 ft/sec

V_{T_y} = 78 ft/sec

V_{T_z} = -300 ft/sec

Table 7

Linear Dynamic Model Vectors

State vector of perturbation variables:

$\bar{x}^T = [\Delta\psi, V_T, W_T, d, \beta_{EL}, \beta_{AZ}, \Theta, \phi, \alpha, q, \beta, p, r]$

$\Delta\psi$ = relative heading $(\psi_T - \psi_A)$

V_T, W_T = target velocities (attacker's coordinates)

d = range variation

β_{EL}, β_{AZ} = target line of sight angle components

Θ, ϕ = attacker Euler angles

α, β = attacker angle of attack, sideslip angle

p,q,r = stability axis angular rates

Pilot's observation vector:

$\bar{y}^T = [\varepsilon_{EL}, \dot{\varepsilon}_{EL}, \varepsilon_{AZ}, \dot{\varepsilon}_{AZ}, \lambda_{EL}, \dot{\lambda}_{EL}, \lambda_{AZ}, \dot{\lambda}_{AZ}, A_y, \phi]$

ε_{EL}, ε_{AZ} = elevation and azimuth tracking errors

λ_{EL}, λ_{AZ} = sight displayed lead angle components

A_y = lateral acceleration

ϕ = relative bank angle

Pilot's control input vector (perturbation deflections):

$\bar{U}_p^T = [\delta_{E_{st}}, \delta_{A_{st}}, \delta_{R_{ped}}]$

$\delta_{E_{st}}$ = elevator deflection

$\delta_{A_{st}}$ = aileron deflection

$\delta_{R_{ped}}$ = rudder deflection

parameters of the baseline model are given in Table 8. Notice that only two control inputs, the aileron and elevator, are included, and the significance of this fact will be discussed later in this section. Simulation data for an F-106 aircraft, with several sight displays, obtained from the large amplitude, motion simulator at the Flight Dynamics Laboratory, Wright-Patterson Air Force Base, was used to calculate rms values for various parameters. The rms performance characteristics of the simulation parameters were then compared to the baseline OCM analytical results, and the results are shown in Table 9. Considering that the baseline optimal control model was essential a "single-axis" model, the values are remarkably close. The system's sensitivities, at least to first order, were then found by first increasing the observation errors, and then also increasing the neuromuscular lag time constant, to determine the changes in rms performance. The results are presented in Table 10, and as expected, all the rms values for the states and outputs increased, but it should be noted that the lateral error increased significantly more than longitudinal error. This tends to indicate the pilot has a greater difficulty reducing lateral errors and must therefore increase the attentional allocation for that axis. Based on rms performance matching, a final model was developed.

The parameters for the final OCM are given in Table 11, and there are several important points to notice. First, there are three control inputs instead of the two control inputs in the baseline model. The rudder was added to account for two different tracking strategies. The first strategy, called the "rolling" strategy, has the pilot first roll the aircraft to null out the azimuth error, using only ailerons, and

Table 8

Baseline OCM

Observation vector, $\bar{y}^T = [\epsilon_{EL}, \dot{\epsilon}_{EL}, \epsilon_{AZ}, \dot{\epsilon}_{AZ}, \lambda_{EL}, \dot{\lambda}_{EL}, \lambda_{AZ}, \dot{\lambda}_{AZ}]$

Objective function weights (both axes), $Q_\epsilon = 16.0$, $Q_{\dot{\epsilon}} = 1.0$, $Q_{\dot{\lambda}} = 4.0$

Observation thresholds, $T_\epsilon = T_\lambda = 0.05$ deg.

$T_{\dot{\epsilon}} = T_{\dot{\lambda}} = 0.10$ deg.

Observation noise ratio, -20 dB for full attention

Fractional attention, $f_i = 0.5$ for all observed variables

Observation delay, $\tau = 0.2$ sec.

Neuromuscular lag, $\tau_N = 0.2$ sec. for all control inputs

Motor noise variance, varied to match rms control inputs

Control inputs, $\delta_{E_{st}}$,$\delta_{A_{st}}$ only

Table 9

Simulation vs. Baseline Model

(RMS Performance)

	ϵ_{EL}(deg)	ϵ_{AZ}(deg)	λ_{EL}(deg)	λ_{AZ}(deg)
Simulation	1.09	0.97	1.72	2.58
Baseline Model	1.31	0.82	0.74	0.34

	V_T(ft/sec)	W_T(ft/sec)	Θ(deg)	ϕ(deg)
Simulation	34.0	48.0	1.09	4.18
Baseline Model	15.0	37.0	0.74	2.29

	p(deg/sec)	q(deg/sec)	r(deg/sec)	α(deg)
Simulation	4.99	1.72	0.92	1.40
Baseline Model	6.07	3.04	1.72	1.03

	$\delta_{E_{st}}$(in)	$\delta_{A_{st}}$(in)	$\delta_{R_{ped}}$(in)	β(deg)
Simulation	0.27	0.22	0.22	0.20
Baseline Model	0.28	0.23	-	0.43

Table 10

OCM Variation Comparison

(RMS Performance)

Model A - Increased observation errors

Model B - Increased observation errors, plus increased neuromuscular lag constant

	ϵ_{EL}(deg)	ϵ_{AZ}(deg)	λ_{EL}(deg)	λ_{AZ}(deg)
Baseline Model	1.31	0.82	0.74	0.34
Model A	1.60	1.32	0.97	0.52
Model B	1.66	1.95	0.34	0.69

	V_T(ft/sec)	W_T(ft/sec)	Θ(deg)	ϕ(deg)
Baseline Model	15.0	37.0	0.74	2.29
Model A	22.0	44.0	0.97	3.32
Model B	29.0	21.0	1.38	4.70

	α(deg)	β(deg)	$\delta_{E_{st}}$(in)	$\delta_{A_{st}}$(in)
Baseline Model	1.03	0.43	0.28	0.23
Model A	1.09	0.47	0.28	0.27
Model B	1.15	0.40	0.24	0.25

	p(deg/sec)	q(deg/sec)	r(deg/sec)
Baseline Model	6.07	3.04	1.72
Model A	6.65	3.09	1.78
Model B	6.30	2.58	1.38

Table 11

Final Multiaxis Pilot Model Parameters

Observation vector, $\bar{y}^T = [\epsilon_{EL}, \dot{\epsilon}_{EL}, \epsilon_{AZ}, \dot{\epsilon}_{AZ}, \lambda_{EL}, \dot{\lambda}_{EL}, \lambda_{AZ}, \dot{\lambda}_{AZ}, A_y, \phi_{rel}]$

Objective function weightings, $Q_{\epsilon} = 16.0$, $Q_{\dot{\epsilon}} = 1.0$, $Q_{\dot{\lambda}} = 4.0$,
$Q_{\phi} = 8.0$, $Q_{Ay} = .007$

Observation thresholds, $T_{\epsilon} = T_{\lambda} = 0.05$ deg
$T_{\dot{\epsilon}} = T_{\dot{\lambda}} = 0.18$ deg
$T_{Ay} = 0.4$ ft/sec^2
$T_{\phi} = 5$ deg.

Fractional attention allocations, fi = 0.05 for $\epsilon_{EL}, \dot{\epsilon}_{EL}, \lambda_{EL}, \dot{\lambda}_{EL}$
fi = 0.15 for $\epsilon_{AZ}, \dot{\epsilon}_{AZ}, \lambda_{AZ}, \dot{\lambda}_{AZ}$
fi = 0.10 for A_y, ϕ

Observation noise ratios, -20dB for full attention

Observation delay, $\tau = 0.2$ sec.

Neuromuscular lag, $\tau_N = 0.33$ for $\delta_{E_{st}} (Q_{\delta_E} = 0.05)$
$\tau_N = 0.23$ for $\delta_{A_{st}} (Q_{\delta_A} = 0.10)$
$\tau_N = 0.62$ for $\delta_{R_{ped}} (Q_{\delta_R} = 0.12)$

Motor noise ratios, $V_n = \pi \rho_u T_u^2$
$\rho_u = .05$(-8dB) for $\delta_{E_{st}}$
$\rho_u = .02$(-12dB) for $\delta_{A_{st}}$
$\rho_u = .05$(-8dB) for $\delta_{R_{ped}}$

then correct the elevation errors, using the elevator, with small azimuth corrections, using the ailerons. A second, or "pointing", strategy could also be used, however, whereby the pilot uses ailerons, elevator, and rudder to continually "point" at the target. This latter strategy requires a considerable amount of "cross controlling" of the control surfaces, however. The simulation results, shown in Table 9, indicate that sideslip, β, is fairly small, and hence a "rolling" strategy was probably employed. From my own personal experiences as an F-4 fighter pilot, I tend to agree that the majority of tracking is done using the "rolling" strategy, but that as the target range gets smaller, or target motion increases, a "pointing" strategy must often be used to maintain the sight on the target. Therefore, the three inputs were included to allow for all possibilities. The second point worthy of notice is the change in the pilot's observation vector, where the lateral acceleration, a_T, and relative bank angle, ϕ_T, have been added. Lateral acceleration was included because test pilots indicated that pilots tend to minimize lateral acceleration, even if larger errors result. Again, from personal experience, I can verify this fact. The inclusion of the relative bank angle seems appropriate, since fighter pilots use target relative bank angle to help predict the target's movement. While the short range tracking solutions simulated in this research tend to keep this angle small, it could be very important for future studies. The analytical performance predictions were then compared to simulation results, and both are shown in Table 12. The double asterisks indicate that the final model provided the best correlation between analytical and simulation results, while the single

Table 12

Final Model Performance (RMS)

	ϵ_{EL}(deg)	ϵ_{AZ}(deg)	λ_{EL}(deg)	λ_{AZ}(deg)
Simulation	1.09	0.97	1.72	2.58
Analytical	1.78	1.72	1.38**	0.63*

	V_T(ft/sec)	W_T(ft/sec)	Θ(deg)	ϕ(deg)
Simulation	34.0	48.0	1.09	4.18
Analytical	25.0*	63.0	0.97*	3.55*

	$\delta_{E_{st}}$(in)	$\delta_{A_{st}}$(in)	$\delta_{R_{ped}}$(in)	α (deg)
Simulation	0.27	0.22	0.22	1.40
Analytical	0.23	0.15	0.10	0.97

	p(deg/sec)	q(deg/sec)	n(deg/sec)	β(deg)
Simulation	4.99	1.72	0.92	0.20
Analytical	5.21**	2.58**	0.57**	0.15**

	τ_{N_E}(sec)	τ_{N_A}(sec)	τ_{N_R}(sec)	A_y(ft/sec^2)
Simulation	-	-	-	0.67
Analytical	.33	.28	.62	0.78**

asterisk means one other model was better. Since only one pilot flew the simulations, and only one simulation run met the required engagement parameters and sight configuration, further correlation would be pointless. Therefore, the final optimal control model will be used as the system model for applying the augmentation control synthesis.

The augmented control synthesis for the multiaxis tracking task follows a similar procedure to the pitch tracking task. Because the "rolling" strategy was assumed to be used, no control gains, however, were put on the rudder. The analytical predictions of rms performance, using the augmented system, are given in Table 13, and a significant reduction in tracking errors, particularly in azimuth errors, is indicated. This indicates that the augmented system makes the largest impact in the areas where the pilot has the greatest difficulty. It should also be noted that the stick and rudder deflections have been reduced, which indicates a reduction in pilot workload. Therefore, the augmentation control synthesis yields a marked improvement, at least analytically, in the piloted vehicle performance. Some small changes will now be added to the optimal control model to more realistically simulate the multiaxis air-to-air tracking task.

2.5 Model Modification

The primary objective of this research is to simulate a multiaxis air-to-air tracking task and use the simulation data to validate the augmentation control synthesis, and if necessary, to change the OCM parameters to better predict the piloted vehicle performance. In an attempt to more realistically simulate the task, two changes have been made to the final OCM. First, the target range is set to a constant

Table 13

Augmented System Performance
(Analytical RMS Results)

	ε_{EL}(deg)	ε_{AZ}(deg)	λ_{EL}(deg)	λ_{AZ}(deg)
Unaugmented	1.78	1.72	1.38	0.63
Augmented	0.54	0.20	0.33	0.10

	V_T(ft/sec)	W_T(ft/sec)	Θ(deg)	ϕ(deg)
Unaugmented	25.0	63.0	0.97	3.55
Augmented	4.0	15.0	0.17	0.48

	$\delta_{E_{st}}$(in)	$\delta_{A_{st}}$(in)	$\delta_{R_{ped}}$(in)	α(deg)
Unaugmented	0.23	0.15	0.10	0.97
Augmented	0.19	0.11	0.06	0.36

	p(deg/sec)	q(deg/sec)	n(deg/sec)	β(deg)
Unaugmented	5.21	2.58	0.57	0.15
Augmented	1.20	0.97	0.21	0.06

	τ_{N_E}(sec)	τ_{N_A}(sec)	τ_{N_R}(sec)	A_y(ft/sec^2)
Unaugmented	0.33	0.28	0.62	0.78
Augmented	0.33	0.28	0.62	0.25

2000 feet. This was done primarily to make the analysis of data easier. The second change introduces perturbations to the constant bank, four-g turn used to initially develop the target motion. Target perturbations are introduced by assuming the pilot will maintain a constant four-g turn, but his bank angle will perturbate about the steady-state value of 75.5°. To determine how these bank angle changes will affect the target motion, the assumptions used to develop the target's motion must be studied. The values of $\dot{v}_T$ and $\dot{w}_T$, in the $\bar{x}$-vector, are the target's velocity perturbations, in the attacker's coordinate system. The equations governing them assumed $\dot{v}_{T_T}$ and $\dot{w}_{T_T}$, the target's velocity perturbations from the steady-state values, in the target's coordinate system, are zero. When the bank angles vary about the steady-state target bank angle, this assumption is no longer true, and $\dot{v}_{T_T}$ and $\dot{w}_{T_T}$ must be added to the $\dot{v}_T$ and $\dot{w}_T$ equations. However, the values of $\dot{v}_{T_T}$ and $\dot{w}_{T_T}$ must be transformed from the target's coordinate system to the attacker's coordinate system before they can be added together. The coordinate transformation to accomplish this is fairly straight forward, and a more detailed development is shown in Appendix B. Now, it is necessary to show how to solve for $\dot{v}_{T_T}$ and $\dot{w}_{T_T}$.

The target motion perturbation equations are given by $\dot{v}_{T_T} = (g \cos \Phi_1)\phi_T$ and $\dot{w}_{T_T} = (-g \sin \Phi_1)\phi_T$, where $\Phi_1 = 75.5°$, the steady-state target bank angle, and ϕ_T = the target perturbation bank angle. Therefore, the problem is to solve for a perturbation bank angle that realistically simulates a real target's motion. First, ϕ_T was assumed to be small, so that q_{T_T} and r_{T_T}, the target perturbation Euler angles, are assumed equal to zero. Therefore, the target perturbations equations are given by $\dot{\psi}_T = 0$, $\dot{\Theta}_T = -.161\ \phi_T$, and $\dot{\phi}_T = p_T + .161\ \Theta_T$.[1,93] The value of p_{T_T} is then calculated using $\dot{p}_T = -\frac{1}{\tau_R} p_T + \delta_A$[7,456], where τ_R is the target vehicle roll time constant, and δ_A is the aileron control surface deflection, δ_A, is calculated by using a sum of sinusoids using $\delta_A(t) = \sum_{i=1}^{13} A_i \sin(w_i t + \phi_i)$, with the variables defined and tabulated in Appendix C. Once the value of δ_A is computed, the target perturbation equations for p_T, Θ_T, and ϕ_T are integrated to determine ϕ_T, which in turn is used to compute $\dot{v}_{T_T}$ and $\dot{w}_{T_T}$. These values are then transformed into the attacker's coordinate system and added to the $\dot{v}_T$ and $\dot{w}_T$ equations, respectively. One last note; the value of τ_R, the target's vehicle roll constant, was set at 2.0, which is the approximate T-38 roll time constant at the given flight conditions. These small changes to the model result in a much more realistic tracking problem.

2.6 *Summary*

The optimal control model (OCM) is found to be an extremely valuable tool for predicting piloted vehicle performance when higher order vehicle dynamics are present. The OCM, once the objective cost function for the particular task is discovered, is a fairly simple and straight forward method of analyzing piloted vehicle performance. The comparison between the analytical and simulation results provides excellent correlation. A systematic augmentation control synthesis procedure for computing the augmentation gains is then applied to the OCM to improve the piloted vehicle performance. The control synthesis has been applied to a pitch tracking task, using aircraft with two sight configurations, whose dynamics result in higher order system dynamics. The analytical predictions of improved piloted vehicle performance, when augmentation is added, must still be verified experimentally, but a vast improvement in performance is theoretically possible. A similar analysis, using the more realistic multiaxis tracking task, shows similar results, but it, too, requires experimental verification, which is the primary objective of this research. To provide for a realistic simulation task, an enhanced degree of target motion has been added to the OCM. The next chapter will provide a detailed analysis of how the multiaxis tracking task will be simulated.

CHAPTER 3. SIMULATION SYSTEM

3.1 Introduction

The Purdue University Aeronautical and Astronautical Engineering (A&AE) Flight Simulation Laboratory provides the basis for simulating the multi-axis air-to-air tracking task. This fixed base simulation was originally developed in 1978,[8] and has been used for several research simulations here at Purdue University. A more complete discussion of the system can be found in Reference 8, but this chapter will first present a brief overview of the hardware components used in the simulation, emphasizing their functions in the simulation, not their specifications. Then, the development of the software to simulate the task will be detailed.

3.2 System Hardware

The hardware used in the simulation can be divided into four main areas: the computer, the display mechanisms, the control input apparatus, and the input-output (I/O) devices. This section will discuss the function and importance of each of these areas.

The heart of any computer simulation is the computer, since it not only must calculate the various parameters used to update the display, but also acts as the command center for controlling the other pieces of hardware. The General Automation SPC-15/45 minicomputer performs these functions and has full I/O capability with the disk, plus a complete range of real time clock functions, which makes the minicomputer well suited for this type of simulation. The minicomputer is also capable of performing I/O operations with several other I/O devices, which will be detailed later in this section. The minicomputer obviously calculates various parameters, which will be fully explained in Section 3.3, but it also acts as the workhorse for updating the graphics display. The GA SPC 15/45 uses a display listing, which is a defined storage location, where various display "entities" are stored. The minicomputer then executes various graphics instructions that individually change or move the various entities, and these binary plotting instructions are then sent to the display mechanisms. The six entities and set of possible graphics instructions used in the simulation are presented in Appendix E.

The display mechanisms consist of two pieces of equipment: the QVEC 2150 vector generator and the Hewlett-Packard 1310A cathode ray tube (CRT). The vector generator is the display processor, which accepts the incoming binary plottings instructions and calculates the positioning voltages and on/off signals used to update the CRT display. The vector generator also has direct access to the display listing contents of the minicomputer, so that the computer execution time for updating the CRT electron beam instructions, which must be done about 30 times per second to prevent flickering, is kept to a minimum.(8) The Hewlett-Packard CRT is an 11"x15" screen with a 50 lines/inch resolution, and the execution time for drawing the beams is essentially zero.(8) These two pieces of hardware combined to form an effective method for displaying a simulation display.

Another important piece of simulation hardware is the apparatus for obtaining the control inputs. The apparatus includes a nonmoving stick to obtain elevator and aileron control inputs and a nonmoving bar to obtain rudder inputs, and is similar to some newer fighter aircraft designs, such as the F-16. When a force is applied to the stick or rudder, resistance strain gages measure a voltage that is proportional to the applied force.(8) The maximum output voltage is ± 10 v, but the amount of force required to obtain the maximum voltage can be changed by varying the control gains, which is done in the software, and will be further explained in Section 3.3. Trim knobs are also included that can adjust the zero-force voltage by ±1 volt. The method of reading these voltages, and then converting them to a control surface deflection, is again accomplished by the software, and will be explained in Section 3.3.

The final hardware component area is input-output devices, and in addition to the disk unit in the minicomputer, includes a card reader, a Superbee teletype terminal, a line printer, and the VERSATEC electrostatic plotter/printer. The minicomputer disk is primarily used to store working computer programs and act as a simulation data storage unit. The data storage on the disk provides an easy method of cataloging the simulation data and still allows an easy method of transferring the data to more permanent storage devices, such as magnetic tape. The elements of various matrices used to update the simulation parameters can be inputted into the computer program by either the card reader or by using a storage location on the minicomputer disk. The latter method was chosen for this simulation because the elements of these matrices are constant. If the elements varied between runs, reading this information on the card reader would be more convenient. The teletype terminal could also be used to input information, but the primary use is for editing programs. The teletype terminals can also be useful for rapidly looking at the simulation output, while the line printer provides a more permanent printed record. The electrostatic printer/plotter is especially useful because the simulation data can be plotted almost immediately. It can also be used to pring the same information as the line printer. A more complete description of how each of these I/O devices are used for this experiment is presented in Chapter 4.

The hardware used for the multiaxis tracking task simulation provides all the equipment to successfully accomplish the simulation. The GA minicomputer, combined with the vector generator and cathode ray tube, calculates the appropriate simulation parameters and then updates the display. The various input/output devices allow for a wide range of inputting information, storing and transferring data, and presenting results. While the hardware is the backbone, or framework, of the simulation, the software is the "brains" of the simulation, and will be discussed next.

3.3 System Software Development

The computer software which accomplishes the multiaxis tracking task simulation can be divided into four phases: the initialization and precomputation phase, the vector update phase, the display update phase, and the data storage and transfer phase. A computer flow chart is shown in Figure 3 and provides a overview of the overall programming sequence. The actual simulation computer listing is given in Appendix F

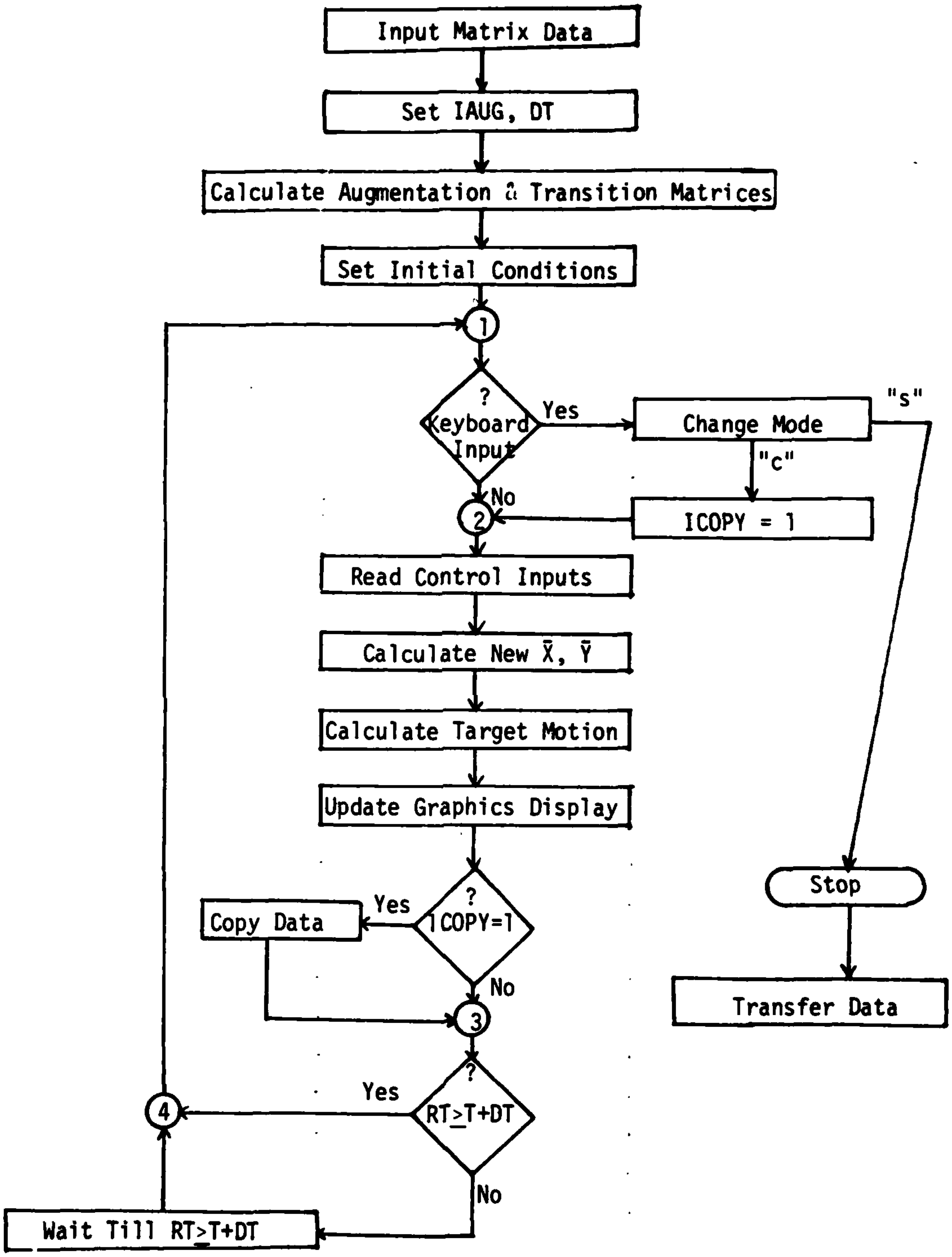


Figure 4

Computer Flow Diagram

and also includes variable definitions and subroutine explanations. This section will outline the development of each phase of the software and give an explanation of why certain methods were chosen.

The initialization and precomputation phase begins by setting up a data file to allow inputting of information through the teletype terminals. The next step is to "read" in the values of the appropriate matrices. Once this is accomplished, the display area is initialized, which means that the name and dimension of the storage area containing the display list is determined. Also, two variables, IAUG and DT, are input via the Superbee teletype terminal. The variable, IAUG, is a logical variable that determines if the augmentation system is included in the simulation run or not. If IAUG = 0, no augmentation is included, while any other value will provide system augmentation. The parameter, DT, is the iteration step size, Δt, and its value and importance will be explained later in this section. These variables are input in this manner to allow for changing the parameters once the program is compiled. The value chosen for DT will be further explained later in this section. The next step in this phase is to calculate the augmentation matrices, A_{aug} and B_{aug}, and the state transition matrices, $e^{A\Delta t}$ and $(\int_0^{\Delta t} e^{A\tau}d\tau)B$, where A and B represent either the augmented or unaugmented system matrices. In the actual programming, only A_{aug} and B_{aug}, defined as AAUG and BAUG, are used, because when IAUG = 0, AAUG and BAUG are set equal to the A and B matrices, respectively, while when 1AUG ≠ 0, AAUG and BAUG are calculated using the control synthesis relationships of $A_{aug} = A\text{-}Bk_x$ and $B_{aug} = B\text{-}Bk_u$. The calculation of the state transition matrices is done by a series approximation and the details are presented in Appendix D. Finally, prior to beginning the iteration cycle, all the initial values are set.

The most important phase of a computer simulation is the vector update phase, where the state and observation vectors are updated at each iteration step, Δt. Before detailing the method used, a brief discussion of why the state transition method was chosen will be given. The two important parameters for updating the state and observation vectors are speed and accuracy. The speed is important, because the display is only updated once during each iteration cycle, and a total cycle time greater than about .1 second will not adequately simulate the task. This is because the minicomputer obtains discrete solutions, which must reasonably represent the aircraft's highest frequency mode, and a value of Δt less than .1 second provides a reasonable speed of display updating.(8) The accuracy is obviously important, because the values of the state and observation vectors are used to update the display. The original A&AE minicomputer system was designed to simulate aircraft motion using six degree of freedom, nonlinear equations of motion. A fourth order Runga-Kutta integration routine was found to have both the required speed and accuracy to meet the original simulation needs.(8) Several other simulations also used this method, even though linear equations of motion were used. Therefore, it seemed reasonable to incorporate the fourth order Runga-Kutta routine for the multiaxis tracking task. A problem arose, however, because the iteration cycle time, Δt, was nearly 0.2 seconds, which was clearly unacceptable. The increase in cycle time resulted from the added complexity of the task, since the previous simulations were primarily pitch tracking tasks, and the state and observation vector dimensions were less than half the dimension of the multiaxis case. Therefore, the state transition method was chosen as a second candidate. The state transition method calculates the new state values using

$$\bar{x}(t+\Delta t) = e^{A\Delta t}\,\bar{x}(t) + \left(\int_0^{\Delta t} e^{A\tau} d\tau\right) B\bar{u}(t)\ . \qquad (3.3.1)$$

This method will calculate exact answers if the inputs, $\bar{u}(t)$, are held constant over the interval, Δt. Since the inputs are only sampled once during each cycle, they are not constant over the interval. However, the fourth order Runga-Kutta method uses the inputs in a similar method, so accuracy of either method seems reasonable. The state transition matrices, $e^{A\Delta t}$ and $(\int_0^{\Delta t} e^{A\tau} d\tau)B$, are constant, once the interval time, Δt, is specified, so they can be calculated prior to the iteration cycle. The savings in iteration computation time was considerable, and the total iteration cycle time was reduced to approximately 0.076 seconds. One drawback is that the cycle time, Δt, cannot vary, and as will be seen later, the data transfer method requires an additional 0.0125 seconds, but only every seventh iteration. Therefore, the Δt for each iteration must account for this added cycle time. Since the overall Δt needs to be greater than 0.0887, the Δt still remains within acceptable limits, so no serious problem results. The state transition method, then, was found to have both the required speed and accuracy, and is used to update the state vector.

The vector update phase uses the control input values to calculate the new values of the state and observation vectors, and then calculates the target motion by integrating the target motion perturbation equations. The control input apparatus was explained in Section 3.2, and a FORTRAN subroutine, DASIN, converts the analog voltages, which are proportional to the applied stick forces, into a digital voltage by averaging four voltage readings over a 1 msec time period. This voltage is then multiplied by a control device gain (in/volt) that converts the stick and rudder pedal forces into control surface deflections (in). The gains were obtained experimentally to yield a reasonable "feel" of the aircraft and set at 1.0, 0.5, and 0.5 for elevator, aileron, and rudder. These control input deflection values make up the $\bar{u}_p$ vector and are used to calculate the new values for the state and observation vectors, using

$$\bar{x}(t+\Delta t) = e^{A\Delta t}\,\bar{x}(t) + \left(\int_0^{\Delta t} e^{A\tau}\,d\tau\right)B\bar{u}(t) \tag{3.3.2}$$

$$\bar{y}(t+\Delta t) = C\bar{x}(t) + D\bar{u}(t)\ . \tag{3.3.3}$$

The next step in the vector update phase is to calculate the target motion. The perturbation equations governing the target motion were developed in Section 2.5 and are repeated here.

$$\dot{p}_T = -\frac{1}{\tau_R}\,p_T + \delta_A \tag{3.3.4}$$

$$\dot{\Theta}_T = -.161\ \phi_T \tag{3.3.5}$$

$$\dot{\phi}_T = p_T + .161\ \Theta_T\ . \tag{3.3.6}$$

The perturbation target aileron deflections, δ_A, were modeled using a sum of sinusoids, where $\delta_A(t) = \sum_{i=1}^{13} A_i \sin(w_i t+\phi_i)$, and the values of A_i, w_i, and ϕ_i are given in Appendix C, and were chosen to simulate some realistic target motion. Once $\delta_A(t)$ is calculated, the perturbation equations are integrated using the fourth order Runga-Kutta integration routine, to obtain ϕ_T, which is used to calculate $\dot{v}_{T_T}$ and $\dot{w}_{T_T}$. As explained in Section 2.5, $\dot{v}_{T_T}$ and $\dot{w}_{T_T}$ must be transformed from the target's coordinate system into the attacker's coordinate system. The actual transformation matrix development is shown in Appendix D, and the actual coding in the computer program was chosen to minimize computation time. Once the state and observation vectors and the target motion have been calculated, the simulation display can be updated.

The next phase of the simulation program, where the simulation display is updated, plays an important part in the simulation, because the pilot's strategy, or selection of control inputs, is based on reacting to the visual display. It is especially important for this system because the pilots only feedback of his control input is through the display. The simulation displays, as depicted in Figure 4, consists of a stationary reference cross, a sight symbol, a target symbol, and three horizon lines. As mentioned in Section 3.2, the minicomputer stores these entities in the display list, and various FORTRAN graphics instructions place them in the proper location on the CRT. The list of entities and the set of possible graphics instructions are contained in Appendix E. The reference cross, which is set when the display area is initialized, represents the attacking aircraft's forward velocity vector, and as mentioned, remains fixed on the screen. The other five entities are moved and rotated on the CRT using the graphics instructions, based on the state and output vectors and target motion values. For example, the sight and target locations are positioned on the CRT using the instruction, GVECT, and the values λ_{EL}, λ_{AZ}, β_{EL}, and β_{AZ}, which are the Y(5), Y(7), X(4), and X(5) values, respectively. In addition, the target symbol is rotated, based on ϕ_T, which was found in the target

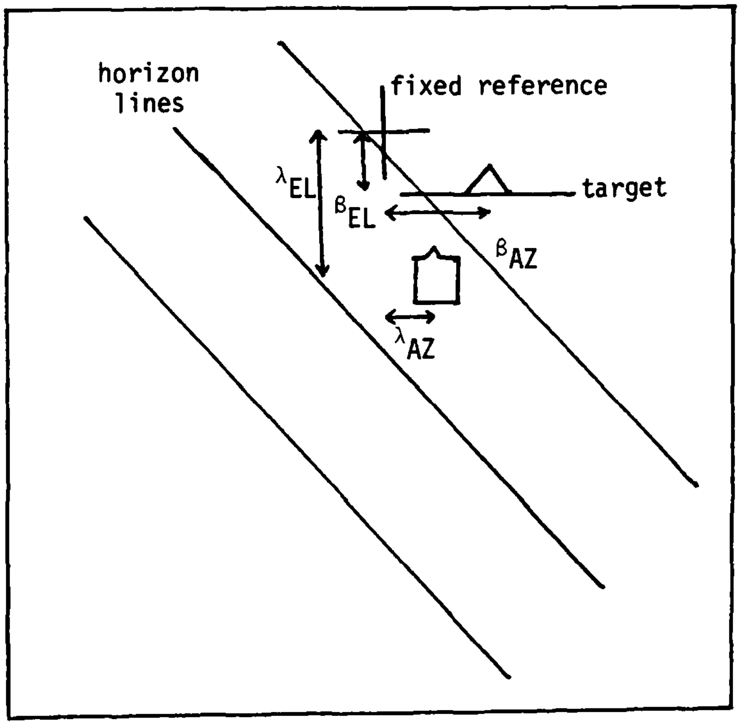


Figure 5

Simulation Display

motion calculations. The values of λ and β are angles, measured in radians, that must be converted into raster units by multiplying them by scale factors, to determine the actual "distance" on the CRT the symbols are located. These scaling factors are somewhat arbitrary, and were chosen as 300.0, based on experimentation, to be able to provide "reasonable" movement of the sight and target. My personal experiences as an F-14 fighter pilot, with included many hours of air-to-air combat flying, were used to make the display as realistic as possible. The horizon lines are included to give the pilot an easy visual cue of his bank angle, which is calculated by adding the attacker's perturbation bank angle, X(7), to 75.5°, the steady-state bank angle. The lines are drawn using trigonometry, and the appropriate graphics instructions, to position three equidistant horizon lines. Once the simulation display has been updated, the data can be stored and transferred.

The final phase of the computer software, which is the data storage and transfer phase, is important not only for future evaluation of the simulation, but also because the method used could possibly interfere with the iteration cycle time. The actual variables that are stored at the end of each iteration cycle are the time, T, the twelve elements in the state vector, $\bar{x}(t)$, the first nine elements of the observation vector, $\bar{y}(t)$, the three control inputs, $\bar{u}_p$, the target's aileron deflection, δ_A, and the target perturbation bank angle, ϕ_T. These values are initially stored in a 28 by 7 storage matrix, XBUF at the end of each iteration. At the end of every seventh iteration, a FORTRAN subroutine, called FSDIO, moves the entire storage matrix to a buffer location, where the disk can store the data while the minicomputer simultaneously continues the simulation computations. The movement of data from the storage location to the buffer location requires approximately 0.0125 seconds, which was mentioned earlier in this section. This method was chosen because it is the fastest method of data transfer available on this system, and because it interferes the least with the simulation computations, which keep iteration cycle time low. The 28x7 matrix is used because FSD10 is capable of transferring 400 words of memory, or 200 values, and smaller amounts of data transfer do not appreciably reduce transfer time. At the end of the simulation run, the data is moved to a labelled storage area, where it is available for data analysis. Also at this time, any additional values can be postcomputed and added to the permanent storage location. For this simulation, the equivalent augmented control inputs are postcalculated. More information about the cataloguing and analysis of data can be found in Section 4.4.

3.4 Summary

The simulation system at Purdue University used to simulate the multiaxis tracking task provides a realistic simulation platform. The GA SPC-15/45 minicomputer is the heart and workhorse of the system and not only calculates the appropriate parameters, but also controls the display mechanisms and input/output devices. The nonmoving control input apparatus uses strain gage measurements to convert control stick and rudder pedal forces into control surface deflections. The software development strived for both accuracy and speed, in order to make the simulation as realistic as possible. The state transition method was chosen to update the state vector for just these reasons. The simulation display updates five display entities, based on the calculated values in the state and observation vectors and the target motion calculations. The combination of the hardware and software provide for a highly realistic simulation of the multiaxis air-to-air tracking task.

CHAPTER 4. EVALUATION PROCESS AND RESULTS

4.1 Introduction

The purpose of any research evaluation is to use some method of determining the validity of the research hypotheses. The primary hypothesis for this research is that the augmentation control synthesis will result in improved piloted vehicle performance, while a secondary hypothesis is that RMS statistics obtained from a simulation can be compared to RMS statistics for an analytical study to determine if any OCM parameters should be changed. The method for testing both hypotheses involves simulating a multiaxis tracking task for both the unaugmented and augmented configurations. Several pilots take part in the simulation so that average RMS statistics can be calculated for various parameters. In addition, the pilot's subjective ratings of the augmentation system will help determine the acceptability of the augmented system. While every researcher hopes the evaluation will substantiate the hypotheses, the process will also help determine the validity of any assumptions and hopefully give some clues towards ways of improving the model. This chapter will first examine some of the model assumptions and then explain the evaluation process. An outline of the simulation and RMS calculation procedures will then be given. Finally, a summary of simulation and analytical results will be tabulated and discussed.

4.2 Assumptions

This section will examine two model assumptions and explain how they were handled in the simulation. The initial assumption says the target perturbation bank angle, ϕ_T, which introduces the target perturbation motion, is a small angle. This assumption was made to simplify the transformation matrix computations. The calculation of ϕ_T is outlined in Section 3.3, and the parameters and methodology was chosen to obtain a mean and standard deviation of approximately 0 and 5 degrees, respectively, for a 90 second simulation run. The actual values for each individual simulation run can be found in Appendix H and the average simulation values were -.38 and 5.25. Therefore, the small angle assumption for ϕ_T is assumed to be valid for the simulation evaluation.

The second, and more important, assumption requires a well trained, highly motivated individual to be accomplishing the task. The parameters of the OCM and the control synthesis are both based on this premise, so any comparison between analytical and simulation results must insure this assumption is true. The initial problem was to find individuals who could, after some minimum amount of training, adequately track the target to meet the well trained assumption. The criteria for determining a well trained pilot was the ability to keep the sight symbol touching some part of the target symbol for at least 90 seconds. If an individual could not attain this proficiency within about a 30 minute initial training period, they were not used for the simulation evaluation. Several individuals tried the task, but only four people were able to meet the criteria. Three of the individuals are U.S. Air Force pilots, and additional information about all four individuals can be found in Appendix H. After this initial training session, several 30 second data runs were made to not only improve proficiency, but to help determine how long each simulation run should be. More information about this will be presented in Section 4.3. By the time the data was collected, each individual could be considered well trained in the task. There is no easy objective criteria for determining a highly motivated individual, but each subject is a graduate student who works in the controls area of the Department of Aeronautics and Astronautics at Purdue University, and therefore understands the importance of doing the best possible job of tracking the target. In addition, the simulation was both interesting and challenging, and each individual wanted to show how well they could track the target. It is therefore assumed that each individual was both well trained and highly motivated. The next step is to explain the evaluation process.

4.3 Evaluation Process

This section will outline the approach for testing the two research hypotheses. The evaluation process involves the selection of certain data parameters and how these parameters test the validity of each hypothesis. The primary hypothesis is that the augmentation control synthesis results in improved piloted vehicle performance. The parameters to test this hypothesis are the elevation and azimuth errors and the control surface deflection rates. The errors will determine whether the performance is better while the rates will show if pilot workload is reduced. The parameters compared are average simulation RMS values and analytical RMS values for both the unaugmented and augmented configuration. Each individual flew each configuration twice and the average RMS values will be the average of all simulations runs for that particular configuration. In addition, pilot comments about the task difficulty, task realism, and any undesirable characteristics were obtained. These comments are used to see if the pilot "feels" the augmentation system is helping to accomplish the task better and to show any tendency that the pilot and augmentation controls are in conflict.

The second hypothesis test involves comparing RMS statistics between analytical and simulation results to determine if any OCM parameters need adjustment. The parameters to be compared are ε_{EL}, ε_{AZ}, V_T, W_T, α, β, λ_{EL}, λ_{AZ}, Θ, ϕ, p, q, r, δ_E, δ_A, δ_R, and A_y . It is hoped that the average simulation RMS statistics are reasonably close to analytical RMS statistics. If they are close, this would indicate the OCM is indeed doing a good job of predicting performance. If all the parameters are grossly off, this could mean either the OCM or the control synthesis is in error, or the simulation is defective. In either case, a closer look at the OCM, the control synthesis, and the simulation would have to be made.

4.4 Simulation and RMS Calculation Procedures

This section will outline the procedures for operating the simulation and analyzing the data, and can be used as a guide for other simulations or as a method of retrieving already stored data. The initial step in the simulation process is to insure that the trim of the control surfaces is set as close to zero as possible. The voltages of each control surface can be displayed on the Superbee teletype terminal by typing $DASCAL, and the voltage values of various control surfaces are displayed. The first three channels represent the elevator, aileron, and rudder trim, respectively, and the appropriate trim button can then be adjusted, as appropriate. From experience, the voltages should be less than .0075 volts. The next step to actually start the simulation sequence by typing $$RWAY, which defines the data file, WAYDAT, which contains the elements of the system matrix, and then starts the actual simulation program, called DISPLY. A message will appear on the terminal, saying that the values of IAUG and DT must be typed on the terminal. Once these values are inputted, the simulation display wlil appear and the simulation will continue until the S button is depressed, which stops the program and then moves the data, if any was taken, to a working file location. To begin copying data, the letter C should be depressed any time after the simulation has begun. When the simulation is done, a message will appear on the terminal screen, which explains how to move the data to a more permanent area. The message says the data can be stored by typing $COPY, A, W3, DS(XXXXXX), where DS(XXXXXX) defines the storage location and the identification scheme is described in Appendix H.

A separate program, WAYPLT, is used to calculate the RMS statistics for any parameter and to plot any two parameters on the VERSATEC plotter, if desired. A more complete description of WAYPLT, plus a computer listing, can be found in Appendix G. The first step is to select the appropriate data file by typing $SI=DS(XXXXXX). Then, some output device lables must be set, namely OM and LO. The label, OM, should be $OM=SY, which prints any output messages on the Superbee teletype terminal, while LO, which determines where the RMS statistics will be printed, can be set to several different output devices. The proper command is $LO=XX, where XX=LP means the line printer, XX=EP means the VERSATEC plotter, and XX=SY means the Superbee terminal. Once the data file and output devices are set, the program is called by typing $WAYPLT on the terminal. A message will appear on the Superbee screen which asks for the values of IX, IY, IPLT, IRMS, and IRATE to be set. These parameters determine which variables should be chosen, whether RMS statistics should be calculated, and if any plots should be generated. Appendix G gives a more detailed explanation of the parameters, while Appendix H gives the variable numbers associated with IX and IY. An example of the RMS statistics printout is shown in Figure 5.

4.5 Results

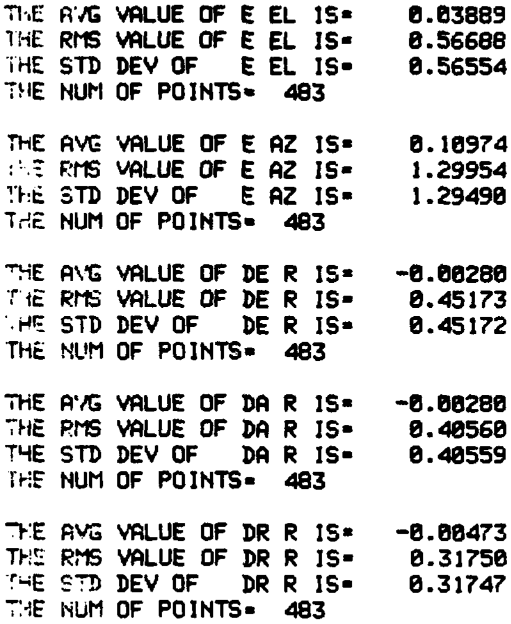
This section contains the tabulation of results and a discussion about them. Tables 14 and 15 give the average RMS statistics of various parameters for both the unaugmented and augmented systems. Table 16 contains a summary of pilot comments. The complete tabulation of RMS statistics and pilot comments for each individual are presented in Appendix H. The information from this section will be the basis for the conclusions given in Chapter 5.

The RMS statistics in Table 14 are for three separate, and different, cases. The analytical statistics were calculated for a target in a constant bank, four-g turn, and the attacker having achieved a steady-state solution. The previous simulation case are the results of the simulation at Wright-Patterson Air Force Base, which was described in Section 2.4, and while the data is based on a target in a constant four-g turn, the pilot did not achieve a steady-state solution, so the

RMS statistics are based on perturbations about some average value. The present simulation results are, of course, based on the target perturbating about the constant four-g turn. Because of these differences, a comparison between the different cases should not result in a one-to-one correlation.

As mentioned in Section 4.3, the RMS statistics are an average of each individual flying each configuration twice. However, for the unaugmented configuration, only two of the four individuals used the rudder, and they had vastly different rudder deflections and rates. By examining the individual statistics, as given in Appendix H, the average value trends is the same as each individual's trend. It is obvious from the comparing these averages values that a one-to-one correlation does not exist, but the differences are primarily due to differing target motions. The addition of extra target banking was expected to result in variations of the RMS statistics, and the largest differences were in parameters associated with the attacker's banking control, such as ε_{AZ}, ω_T, ϕ, p, r, and δ_A. A comparison between the unaugmented and augmented configurations shows that all the parameters have a reduction in average RMS statistics except W_T and λ_{EL}, which have very slight increases.

The pilot comments were based on the 1-10 rating scale mentioned earlier, where the rating of 10, or uncontrollable, meant the individual could not keep the sight symbol touching some part of the target symbol throughout the simulation run. All of the individuals rated the simulation about the same, so the summary contained in Table 16 provides an accurate record of their comments.



```
THE AVG VALUE OF E EL IS=    0.03889
THE RMS VALUE OF E EL IS=    0.56688
THE STD DEV OF   E EL IS=    0.56554
THE NUM OF POINTS=  483

THE AVG VALUE OF E AZ IS=    0.10974
THE RMS VALUE OF E AZ IS=    1.29954
THE STD DEV OF   E AZ IS=    1.29490
THE NUM OF POINTS=  483

THE AVG VALUE OF DE R IS=   -0.00280
THE RMS VALUE OF DE R IS=    0.45173
THE STD DEV OF   DE R IS=    0.45172
THE NUM OF POINTS=  483

THE AVG VALUE OF DA R IS=   -0.00280
THE RMS VALUE OF DA R IS=    0.40560
THE STD DEV OF   DA R IS=    0.40559
THE NUM OF POINTS=  483

THE AVG VALUE OF DR R IS=   -0.00473
THE RMS VALUE OF DR R IS=    0.31750
THE STD DEV OF   DR R IS=    0.31747
THE NUM OF POINTS=  483
```

Figure 6

Sample RMS Statistics Printout

Table 14

Unaugmented System Comparison
(RMS Performance)

PARAMETER	ANALYTIC RESULTS	PREVIOUS SIM RESULTS*	PRESENT SIM RESULTS (AVG)
ε_{EL} (deg)	1.78	1.09	1.33
ε_{AZ} (deg)	1.72	.97	3.80
V_T (ft/sec)	25.0	34.0	18.24
W_T (ft/sec)	63.0	48.0	23.05
α (deg)	.97	1.4	.69
β (deg)	.15	.20	.42
λ_{EL} (deg)	1.38	1.72	.63
λ_{AZ} (deg)	.63	2.58	.52
Θ (deg)	.97	1.09	10.03
ϕ (deg)	3.55	4.18	11.06
p (deg/sec)	5.21	4.99	10.02
q (deg/sec)	2.58	1.72	1.82
r (deg/sec)	.57	.92	1.53
δ_E (in)	.23	.27	.20
δ_A (in)	.15	.22	.52
δ_R (in)	.10	.22	.20
$\dot{\delta}_E$ (in/sec)	-	-	.87
$\dot{\delta}_A$ (in/sec)	-	-	2.22
$\dot{\delta}_R$ (in/sec)	-	-	.88
A_y (ft/sec^2)	.78	.67	2.46
ϕ_T(AVG/RMS)	0.0/5.25	-	-.39/5.22

*Simulation done at Wright-Patterson AFB

Table 15

Augmented System Comparison

(RMS Performance)

PARAMETER	ANALYTIC RESULTS	PRESENT SIM RESULTS (AVG)
ε_{EL}(deg)	.54	.49
ε_{AZ}(deg)	.20	1.31
V_T(ft/sec)	4.0	5.95
W_T(ft/sec)	15.0	24.58
α (deg)	.36	.57
β (deg)	.06	.35
λ_{EL}(deg)	.33	.72
λ_{AZ}(deg)	.10	.17
θ (deg)	.17	9.67
ϕ (deg)	.48	10.03
p(deg/sec)	1.20	2.58
q(deg/sec)	.97	1.15
r(deg/sec)	.21	.34
δ_E(in)	.19	.20
δ_A(in)	.11	.18
δ_R(in)	.06	.33
$\dot{\delta}_E$(in/sec)	-	.36
$\dot{\delta}_A$(in/sec)	-	.38
$\dot{\delta}_R$(in/sec)	-	.58
A_y(ft/sec^2)	.25	1.87
ϕ_T(AVG/RMS)	0.0/5.25	-.38/5.27

Table 16

Summary of Pilot Comments

Was the augmentation system helpful? All subjects said Yes.

Any undesirable or annoying characteristics? None of the subjects mentioned any.

How would you rate the task difficulty from 1-10, where

1 = no inputs required

5 = reasonable difficulty for task

10 = uncontrollable

Unaugmented? Subjects rated between 5 and 7

Augmented? Subjects rated between 2 and 3

4.6 Summary

This chapter has outlined the evaluation process used to test the two research hypotheses. The model assumptions were discussed and it was shown that they were valid for the simulation. The actual process used to evaluate each research hypothesis was then explained. The procedures for running the simulation and calculating the RMS statistics was presented next. Finally, the results of the simulation were tabulated and discussed.

CHAPTER 5. CONCLUSIONS AND SUGGESTIONS

5.1 Introduction

This section will use the results presented in Section 4.4 to draw some conclusions about the research hypotheses. The simulation results show that the control synthesis provides improved piloted vehicle performance. Also, comparing the simulation RMS statistics to analytical results indicates several parameters are considerably different. However, the largest differences are in parameters associated with the attacker's bank control, and since target motion was added to the optimal pilot control model via target perturbation banking, this suggests a logical reason for the disparity. To test this premise, the first suggestion for future research is to do the simulation with no target perturbations and see if the RMS statistics yield better correlation between simulation and analytical results. Another suggestion is to simulate the pitch only tracking task that was described in Section 2.3.

5.2 Conclusions

The primary objective in this research is to determine if the augmentation control synthesis results in improved piloted vehicle performance. It is expected that the RMS statistics will show a decrease in elevation and azimuth errors, since the methodology requires it. However, it is hoped that pilot workload will also be decreased, and the pilot workload was measured objectively by using RMS statistics on the control surface deflection rates, as well as subjectively, based on pilot comments. The average RMS statistics for both the augmented and unaugmented configuration are shown in Table 16, and the elevation and azimuth errors are both reduced, especially in azimuth. The deflection rates for the aileron and elevator are reduced, as well as for the rudder. The lower deflection rates show a reduced piloted workload and this premise is supported by the favorable pilot comments, which were summarized in Table 15. All the pilots rated the task difficulty significantly easier in the augmented case and they all felt that there were no unfavorable characteristics for the augmented system. Because of the lower RMS statistics and favorable pilot comments, it is felt that the piloted vehicle performance is improved significantly by the augmentation system.

The secondary research objective was to compare the analytical RMS statistics of the OCM to simulation statistics in order to determine if any significant parameter changes are required. The analytic results are based on a constant target bank angle, so it is not surprising that the attacker bank parameters do not match very well. Also, note that λ_{AZ}, the sight azimuth angle, shows excellent agreement, and since $\varepsilon_{AZ} = \lambda_{AZ} - \beta_{AZ}$, the disparity in ε_{AZ} must be primarily due to β_{AZ}, the target azimuth angle, and this indicates the target bank is largely responsible for the ε_{AZ} disparity. It is therefore felt that any decision about OCM parameter adjustment should use a similar analytical study. Some observations, however, can still be made about the parameters. It should be noted that the parameters associated with aileron and rudder deflections and deflection rates are considerably higher than the elevator parameters, which indicates the pilots have a much more difficult time of controlling the azimuth error. Also the sideslip angle, β, and lateral acceleration, Ay, are clues that would be available in a motion simulator, and since this simulation is fixed based, it is not surprising that the simulation results are much larger than analytic results. Finally, pilot comments from DJB and WAY, both who have air-to-air tracking experience, indicate the simulation is fairly realistic of actual multiaxis tracking tasks. It is therefore felt that the simulation is realistic, and while the RMS statistics do not match exactly, the introduction of the target perturbation angle accounts for the disparity.

5.3 Suggestions

This section contains three suggestions for possible follow-on research. The first suggestion is fairly obvious and relatively simple to implement. An analytical study with the additional target motion added could be done and the calculated RMS statistics could then be compared to the present simulation results. A second suggestion is to simulate the pitch-only tracking task described in Section 2.3. This type of simulation could be used to determine if a pilot's strategy for correcting elevation would be the same for both a pitch-only and a multiaxis tracking task. A third possibility would be to use some more advanced simulation facility, such as the Flight Dynamics Laboratory at Wright-Patterson AFB to incorporate motion cues.

Table 17

Unaugmented vs. Augmented System Comparison

(RMS Performance)

PARAMETER	UNAUGMENTED	AUGMENTED
ε_{EL}	1.33	.49
ε_{AZ}	3.80	1.31
$\dot{\delta}_E$	.87	.36
$\dot{\delta}_A$	2.22	.38
$\dot{\delta}_R$	.88	.58

5.4 Summary

This chapter seems to validate the decision to use the optimal pilot control model for modeling the multiaxis air-to-air tracking task. The simulation results, as shown by a reduction in elevation and azimuth errors, as well as reduced control surface deflection rates, indicate the piloted vehicle performance characteristics are significantly improved by using the augmentation control synthesis. Pilot comments about the piloted vehicle characteristics also support this idea. While the RMS statistics do not match the analytic statistics, the introduction of target perturbation motion accounts for the disparities. One suggestion for future research is to delete the target motion and then compare simulation results to analytic results. Other suggestions include simulating the pitch tracking task and simulating the multiaxis task on a more sophisticated simulation system.

It should be remembered that this research is not an end in itself, but only one small step in a larger plan. The ultimate goal is to find some theoretical method of accurately predicting piloted vehicle performance, especially when higher order system dynamics are present. Since classical methods are inadequate for accurately predicting the piloted vehicle performance in this situation, the optimal control model was chosen as one method for predicting the performance characteristics. An augmentation control synthesis method was then employed to achieve improved analytic performance, but additional simulation results were required to validate this hypothesis. The primary goal of this research was to simulate a multiaxis air-to-air tracking task, whose analytical development had previously been done by Schmidt.(1)(2) A secondary objective was to use these simulation results to determine if any changes to the OCM parameters were required to better model the task. The simulation RMS statistics, as well as pilot comments, indicate the augmentation synthesis does indeed provide improved piloted vehicle performance. In addition, although the simulation statistics do not closely match all the analytic statistics, the addition of target perturbation motion explains the differences. Therefore, the optimal pilot control model seems one promising method for accurately predicting piloted vehicle performance characteristics and the augmentation control synthesis provides one method for improving the piloted vehicle performance.

LIST OF REFERENCES

LIST OF REFERENCES

1. Schmidt, D.K., "Multivariable Closed-Loop Analysis and Flight Control Synthesis for Air-to-Air Tracking", Final Report for the Air Force Office of Scientific Research, June 1980.

2. Schmidt, D.K., "Pilot-Optimal Augmentation for the Air-to-Air Tracking Task", Journal of Guidance and Control, Vol. 3, No. 5, September-October 1980.

3. Harvey, T.R., "Application of an Optimal Control Pilot Model to Air-to-Air Combat", AFIT Masters Thesis, GA/MA/74, Wright-Patterson AFB, Ohio, March 1974.

4. Guthrie, G.R., "GSINF-A General Introduction to Graphics on the A&AE Computer Facility", unpublished, Purdue University School of Aeronautics and Astronautics, 1978.

5. Kleinman, D.L., Baron, S., and Levinson, W.H., "An Optimal Control Model of Human Response", Parts I and II, Automatica, Vol. 6, 1970, pp. 357-383.

6. Levinson, W.H., et.al., "Studies of Multivariable Manual Control Systems: A Model for Task Interference", NASA CR-1746, May 1971.

7. Roskam, J., Airplane Flight Dynamics and Automated Flight Controls, Roskam Aviation and Engineering Corporation, Lawrence, Kansas, 1979.

8. Jones, B.W. and Schmidt, D.K., "FLTSIM: Flight Simulator, A&AE Minicomputer System", A&AE Departmental Document #78-53, School of Aeronautics and Astronautics, Purdue University, August 1978.

APPENDICES

APPENDIX A

State System Variable Matrices

This appendix contains printouts of all the appropriate matrices. The A, B, C, and D matrices are the linear system matrices defined by

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}$$

$$\bar{y}(t+\Delta t) = C\bar{x} + D\bar{u}$$

The Kx and Ku matrices are the gain matrices calculated by the control synthesis, and the AAUG and BAUG matrices were calculated using

$$AAUG = A - BK_x$$

$$BAUG = B - BK_u \quad .$$

The EXP TAU matrix is

$$\int_0^{\Delta t} e^{A\tau} d\tau$$

and EXP TAU times B matrix is

$$(\int_0^{\Delta t} e^{A\tau} d\tau)B \ .$$

The transition matrix is $e^{A\Delta t}$.

```
                                  THE A MATRIX

0.      0.      0.      0.      0.      0.      0.      0.      -0.9860  0.       0.      -0.2500
0.      0.      0.      0.      0.      -48.6200-110.8000 0.    0.       0.      -302.0000-710.0000
0.      0.      0.      0.      0.      -12.5900 -29.2500 0.    710.0000 0.      -78.0000  0.
0.      0.      0.0005  -0.0063 0.0100  0.      0.      -0.3875  1.0000  0.       0.0387  -0.0100
0.      0.0050  -0.0000 0.0100  -0.0063 0.      0.      -0.0039  0.      -0.3875  -0.1290  -1.0000
0.      0.      0.      0.      0.      -0.0010 -0.1610  0.       0.2500  0.       0.      -0.9680
0.      0.      0.      0.      0.      0.1610  0.      0.       0.      0.       1.0000   0.
0.      0.      0.      0.      0.      0.      -0.0402 -1.2720  1.0000  0.       0.       0.
0.      0.      0.      0.      0.      0.      0.0109 -11.2100 -0.5470  0.       0.0350  -0.0160
0.      0.      0.      0.      0.      0.      0.0104  0.       0.      -0.2280   0.0023  -0.9890
0.      0.      0.      0.      0.      0.      0.      0.      -0.0245 -77.9400   0.1980   5.2700
0.      0.      0.      0.      0.      0.      0.      0.       0.0127  22.1300  -0.3440  -1.8500


          THE B MATRIX

0.          0.          0.
0.          0.          0.
0.          0.          0.
0.          0.          0.
0.          0.          0.
0.          0.          0.
0.          0.          0.
-0.0173000  0.          0.
-0.6060000  0.          0.
0.          0.0028900   0.0034700
0.          -1.8429999  0.6080000
0.          0.2010000   -0.4130000
```

THE C MATRIX

```
 0.0057   0.       0.0004   1.0000   0.       -0.0013  0.       -0.2480  0.       0.       0.       0.
 0.       0.       0.0005  -0.0063   0.0100   -0.0049 -0.0020   -0.0775  1.5060   0.       0.0087  -0.0100
-0.0008   0.0004  -0.0000   0.       1.0000   -0.0075 -0.0183    0.0031  0.      -0.2490   0.       0.
 0.       0.0005  -0.0000   0.0100  -0.0063   -0.0190 -0.0460    0.0009  0.0002  -0.4450  -0.2480  -1.0370
-0.0057   0.      -0.0004   0.0018   0.        0.0013  0.        0.2480  0.       0.       0.       0.
 0.       0.       0.       0.       0.        0.0049  0.0020   -0.3100 -0.5060   0.       0.0300   0.
 0.0008  -0.0004   0.0000   0.       0.0018    0.0075  0.0183   -0.0031  0.       0.2490   0.       0.
 0.       0.       0.       0.       0.        0.0190  0.0460    0.0030 -0.0002  -0.0580   0.1190   0.0370
 0.       0.       0.       0.       0.        0.      0.        0.      0.0943-225.0000  -3.5500  -9.8300
 0.       0.       0.       0.       0.        0.      1.0000    0.      0.       0.       0.       0.
```

THE D MATRIX

```
 0.          0.          0.
 0.0043000   0.          0.
 0.          0.          0.
-0.0000528  -0.0007380  -0.0016400
 0.          0.          0.
-0.0043000   0.          0.
 0.          0.          0.
 0.0000528   0.0007380   0.0016400
 0.          0.          0.
 0.          0.          0.
```

```
                THE KX AUGMENTATION MATRIX

  -0.0330000  -0.0050000  -0.0030000
  -0.0004000   0.0090000   0.0050000
  -0.0040000  -0.0000400  -0.0000200
  -5.3109998   0.1270000   0.0750000
  -0.1480000   5.4699997   3.5849997
   0.0230000  -0.1050000  -0.0810000
  -0.1170000  -1.9770000  -0.3340000
   5.3119997  -0.0310000  -0.0260000
  -0.1970000   0.0440000  -0.0010000
   0.3430000  -9.4849996   7.5169997
   0.0230000  -1.8380001  -0.0479000
   0.0330000  -2.2059998  -5.7009992

                THE KU AUGMENTATION MATRIX

   0.3180000  -0.0020000   0.
  -0.0010000   0.3720000   0.
  -0.0010000   0.0560000   0.
```

```
                                       THE AAUG MATRIX

0.        0.        0.        0.        0.        0.        0.        0.       -0.9860   0.        0.       -0.2500
0.        0.        0.        0.        0.      -48.6200-110.8000   0.        0.        0.     -302.0000-710.0000
0.        0.        0.        0.        0.      -12.5900 -29.2500   0.      710.0000   0.      -78.0000   0.
0.        0.        0.0005   -0.0063    0.0100   0.        0.       -0.3875    1.0000   0.        0.0387   -0.0100
0.        0.0050   -0.0000    0.0100   -0.0063   0.        0.       -0.0039    0.      -0.3875   -0.1290   -1.0000
0.        0.        0.        0.        0.       -0.0010   -0.1610   0.        0.2500   0.        0.       -0.9680
0.        0.        0.        0.        0.        0.1610   0.        0.        0.       0.        1.0000   0.
-0.0006  -0.0000   -0.0001   -0.1005   -0.0026    0.0004   -0.0422   -1.1697    0.9447   0.0059    0.0004    0.0006
-0.0266  -0.0003   -0.0032   -4.6837   -0.1193    0.0185   -0.0834   -6.4449   -3.1238   0.2765    0.0535    0.0106
0.0003   -0.0000    0.0000   -0.0006   -0.0282    0.0006    0.0173    0.0002   -0.0001  -0.2267    0.0077   -0.9628
-0.0074   0.0135   -0.0001    0.1885    7.9015   -0.1443   -3.4405   -0.0413    0.0572 -99.9912   -3.1603    4.6705
-0.0002   0.0003   -0.0000    0.0054    0.3811   -0.0123    0.2594   -0.0045    0.0034  27.1410    0.0057   -3.7611

                THE BAUG MATRIX

0.              0.              0.
0.              0.              0.
0.              0.              0.
0.              0.              0.
0.              0.              0.
0.              0.              0.
1.              0.              0.
-0.0117986     -0.0000346       0.
-0.5456920     -0.0016120       0.
0.0000064       0.0016206       0.0034700
-0.0012350     -1.1914520       0.6080000
-0.0002120      0.1493560      -0.4130000
```

```
                                  THE EXP TAU MATRIX

 0.9012   0.0000   0.0002   0.2113  -0.0018  -0.0014  -0.0010   0.2030  -0.1007  -0.1661  -0.0027  -0.0180
 0.1394   0.4661   0.0018  -2.3207-147.0743 -11.4768  10.0902   2.5879  -0.9368 192.3979 -34.0335-123.5235
-5.8349  -0.1147   0.7739-152.7827 -36.7624  -1.9980   3.1599-145.4992  72.2827 172.2131  -6.8801 -18.8309
-0.0012   0.0000   0.0000   0.6830   0.0147  -0.0004  -0.0069  -0.3243   0.1065  -0.0742   0.0037   0.0104
 0.0002   0.0013   0.0000  -0.0008   0.6339  -0.0168   0.0187   0.0022  -0.0015  -0.1403  -0.0931  -0.3309
-0.0093  -0.0001  -0.0000  -0.0540  -0.0275   0.8979  -0.0670  -0.0510   0.0253  -0.5894  -0.0087  -0.0622
-0.9204   0.0013  -0.0000   0.0068   0.4233   0.0334   0.7132  -0.0075   0.0027  -2.0770   0.0914   0.2567
-0.0010  -0.0000  -0.0001  -0.1713  -0.0040   0.0008  -0.0115   0.4286   0.0527   0.0111   0.0003   0.0015
-0.9926  -0.0000  -0.0004  -0.4559   0.0066   0.0038  -0.0006  -0.3346   0.0472  -0.0490   0.0048   0.0149
 0.0000  -0.0000   0.0000  -0.0001  -0.0101   0.0005  -0.0048   0.0002  -0.0000   0.1323   0.0012  -0.0343
-0.0068   0.0035  -0.0000   0.0209   0.8799  -0.1094  -0.4196  -0.0100  -0.0016  -1.7671  -0.0285   0.1083
-0.0000   0.0000  -0.0000   0.0001   0.0128   0.0003   0.0176  -0.0002   0.0000   1.0265   0.0036  -0.0081


  THE EXP TAU TIMES B MATRIX

  0.0530   0.0004   0.0052
  0.5539  22.4135  30.9905
-38.0029   5.5524   4.1917
 -0.0547  -0.0031  -0.0023
  0.0010   0.0612   0.0796
 -0.0133   0.0000   0.0184
 -0.0016  -0.0739  -0.0577
 -0.0340  -0.0003  -0.0004
 -0.0220  -0.0036  -0.0034
  0.0000  -0.0063   0.0153
  0.0010   0.0473  -0.0682
  0.0000  -0.0038   0.0091
```

THE TRANSITION MATRIX

```
 1.0026 -0.0000  0.0004  0.4495 -0.0098 -0.0038 -0.0038  0.3300 -0.0466 -0.2083 -0.0056 -0.0127
 2.3087 -0.2361  0.0046 -4.5361-320.4220 -14.5189 38.4865  6.4777 -1.0511 63.6361 -3.6410 -52.4052
-1.7800 -0.3116  0.7013-324.8331 -75.9513 -1.0490 12.2674-235.9170 33.2604 171.1888  3.0424 -4.5884
-0.0027  0.0001  0.0000  0.5306  0.0300 -0.0019 -0.0108 -0.5718  0.0622 -0.0468 -0.0009  0.0058
 0.0008  0.0018  0.0000 -0.0061  0.1385 -0.0436  0.0887  0.0102 -0.0040  0.1114 -0.1699 -0.6080
-0.0006 -0.0003 -0.0001 -0.1151 -0.0789  0.9944 -0.1319 -0.0822  0.0113 -0.6709 -0.0170 -0.0297
-0.0009  0.0035 -0.0000  0.0122  0.8755  0.0352  0.5696 -0.0182  0.0025 -1.8620 -0.0299  0.0983
-0.0016 -0.0000 -0.0003 -0.2888 -0.0061  0.0019 -0.0171  0.2254 -0.0328  0.0235  0.0008  0.0033
-0.0011  0.0001 -0.0004 -0.1836  0.0353  0.0051  0.0087  0.2633 -0.2155 -0.0517  0.0011  0.0091
 0.0000 -0.0000 -0.0000 -0.0002 -0.0075  0.0009 -0.0078  0.0001  0.0001 -0.0726  0.0017  0.0355
 0.0292  0.0041  0.0000  0.0135 -0.1394 -0.2368 -0.2760  0.0108 -0.0152  5.8528 -0.5180 -2.1129
 0.0000  0.0001 -0.0000  0.0001 -0.0037  0.0006 -0.0022  0.0002 -0.0000 -0.8167 -0.0024  0.0109
```

APPENDIX B

State Transition Matrices Calculations

Two state transition matrices, $e^{A\Delta t}$ and $(\int_0^{\Delta t} e^{A\tau})B$, must be precomputed to update the x-vector. The series representation of $e^{A\tau}$ is given by

$$e^{A\tau} = I + A\tau + \frac{1}{2!} A^2\tau^2 + \frac{1}{3!} A^3\tau^3 + \ldots . \qquad \text{(A1-1)}$$

Integrating term by term, over the limits 0 to Δt, yields

$$\int_0^{\Delta t} e^{A\tau} d\tau = I\Delta t + \frac{1}{2!} A(\Delta t)^2 + \frac{1}{3!} A^2(\Delta t)^3 + \ldots . \qquad \text{(A1-2)}$$

Note that multiplying (A1-2) by the matrix, A, and adding the identity matrix, I, gives

$$I + A\Delta t + \frac{1}{2!} A^2(\Delta t)^2 + \frac{1}{3!} A^3(\Delta t)^3 + \ldots \qquad \text{(A1-3)}$$

which is the series representation of $e^{A\Delta t}$. Therefore, the only matrix that must actually be computed is $\int_0^{\Delta t} d\tau$. Then multiply it by A and add I to get $e^{A\Delta t}$, and multiply $\int_0^{\Delta t} e^{A\tau} d\tau$ by the B matrix to obtain the second state transition matrix.

The matrix, $\int^{\Delta t} e^{A\tau}dt$, was computed by adding additional terms of the series until the contribution from the higher order terms was negligible. The first term in the series was called E1, and the second term, E2. If the difference of every element of the resultant matrix is less a specified value, .000001 in this program, then the lower term is added to $\int_0^{\Delta t} e^{A\tau}d\tau$, and the computation is complete. If any element is greater than the specified amount, the lower term is added to $\int_0^{\Delta t} e^{A\tau}d\tau$ and the matrix E1 is set equal to E2. The next higher term is then computed by multiplying the new E1 matrix by $A\Delta t$ and then dividing by the order of the term. For example, the third term in the series is obtained by multiplying the second term by $\frac{1}{3} A\Delta t$. This higher term is now the new E2 matrix and the procedure outlined above is repeated until the higher term produces no appreciable change. The accuracy of these calculations is limited only by computer accuracy, and since they are precomputed matrices, they can be calculated on any computer, to achieve the desired level of accuracy. As mentioned earlier, the two required state transitions are now obtained by multiplying $\int_0^{\Delta t} e^{A\tau}d\tau$ by A and adding the identity matrix and by multiplying $\int_0^{\Delta t} e^{A\tau}d\tau$ by the B matrix.

APPENDIX C

Sum of Sinusoids Control Inputs

The equation used to obtain the target's perturbation aileron deflection is given by

$$\delta A(t) = \sum_{i=1}^{13} A_i \sin(\omega_i t + \phi_i) \qquad \text{(C1-1)}$$

where the values of A_i, ω_i, and ϕ_i are shown below.

i	A_i	ω_i(rad/sec)	ϕ_i(rad)	# of cycles in 100 sec. run
1	0.001	.18850	$\pi/6.5$	3
2	0.001	.31416	$2\phi_1$	5
3	0.001	.50265	$3\phi_1$	8
4	0.001	.87965	$4\phi_1$	14
5	0.05	1.44513	$5\phi_1$	23
6	0.05	2.31628	$6\phi_1$	34
7	0.1	3.07876	$7\phi_1$	49
8	0.1	4.20973	$8\phi_1$	67
9	0.1	5.78053	$9\phi_1$	92
10	0.1	8.23097	$10\phi_1$	131
11	0.1	11.2469	$11\phi_1$	179
12	0.1	15.77079	$12\phi_1$	251
13	0.1	23.93894	$13\phi_1$	381

The values for ω_i, the frequencies, were selected so that no frequency was an integer multiple of any of the other twelve frequencies. The magnitudes, A_i, were selected by experimentation to give an rms value of ϕ_T, using the target motion equations, of approximately 5 degrees. The magnitudes selected above resulted in an rms value of $\phi_T \cong 5.25$ for a $\Delta t = .0887$ and a total time of 90 seconds.

APPENDIX D

Coordinate Transformation Calculations

The transformation from the target's coordinate system ($\bar{i}_A$, $\bar{j}_A$, $\bar{k}_A$), to the attacker's coordinate system ($\bar{i}_A$, $\bar{j}_A$, $\bar{k}_A$), is obtained using the transformation matrix relationships of

$$\begin{pmatrix} \bar{i}_A \\ \bar{j}_A \\ \bar{k}_A \end{pmatrix} = T(-\phi_A)\ T(-\Theta_A)\ T(-\psi_A)\ T(\psi_T)\ T(\Theta_T)\ T(\phi_T) \begin{pmatrix} \bar{i}_T \\ \bar{j}_T \\ \bar{k}_T \end{pmatrix} \quad (1,\ 87) \qquad \text{(B1-1)}$$

where the individual transformation matrices are

$$T(\Theta) = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta \\ 0 & 1 & 0 \\ -\sin\Theta & 0 & \cos\Theta \end{bmatrix} \qquad T(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \ . \qquad \text{(B1-2)}$$

The resultant transformation matrix is represented by

$$\begin{pmatrix} \bar{i}_A \\ \bar{j}_A \\ \bar{k}_A \end{pmatrix} = \begin{bmatrix} T11 & T12 & T13 \\ T21 & T22 & T23 \\ T31 & T32 & T33 \end{bmatrix} \begin{pmatrix} \bar{i}_T \\ \bar{j}_T \\ \bar{k}_T \end{pmatrix} \qquad \text{(B1-3)}$$

One simplification is obtained by remembering that $T(-\psi_A)\ T(\psi_T) = T(-(\psi_A-\psi_T)) = T(-\Delta\psi)$. Also, the value of ϕ_T is equal to $\Phi_1 + \phi_T'$, where Φ_1 = the steady state target bank angle, which is 75.5°, while ϕ_T' is the target's perturbation bank angle. Substituting the values into the appropriate matrices, and using a small angle approximation for ϕ_T' yields:

$$T11 = \cos(\Theta_A)\ \cos(\Delta\psi)\ \cos(\Theta_T) + \sin(\Theta_A)\ \sin\ (\Theta_T)$$

$$T12 = \cos(\Theta_A)\ \cos(\Delta\psi)\sin(\Theta_T)]\sin(\Phi_1) + \phi_T'\ \cos(\Phi_1)] + \cos(\Theta_A)\ \sin(\Delta\psi) .$$

$$[\cos(\Phi_1) - \phi_T'\ \sin(\Phi_1)] - \sin(\Theta_A\ )\ \cos(\Theta_T)[\sin(\Phi_1) + \phi_T'\ \cos(\ \Phi_1)]$$

$$T13 = \cos(\Theta_A)\ \cos(\Delta\psi)\ \sin(\Theta_T)]\cos(\Phi_1) - \phi_T'\ \sin(\Phi_1)] - \cos(\Theta_A)\ .$$

$$\sin(\Delta\psi)[\sin(\Phi_1) - \phi_T'\cos(\Phi_1)] - \sin(\Theta_A)\ \cos(\Theta_T)[\cos(\Phi_1)\ - \phi_T'\sin(\Phi_1)]$$

$$T_{21} = \cos(\Theta_T)[\sin(\phi_A)\ \sin(\Theta_A)\ \cos(\Delta\psi) - \cos(\phi_A)\ \sin(\Delta\psi)] - \sin(\phi_A)\cos(\Theta_A)\ .$$

$$\sin(\Theta_T)$$

$$T22 = \sin(\Theta_T)[\sin(\phi_A)\ \sin(\Theta_A)\ \cos(\Delta\psi) - \cos(\phi_A)\ \sin(\Delta\psi)][\sin(\Phi_1) +$$

$$\phi_T'\cos(\Phi_1)] + [\sin(\phi_A)\ \sin(\Theta_A)\ \sin(\Delta\psi) + \cos(\phi_A)\ \cos(\Delta\psi)]\ .$$

$$[\cos(\Phi_1) - \phi_T'\ \sin(\Phi_1)] + \sin(\phi_A)\ \cos(\Theta_A)\ \cos(\Theta_T)]\sin(\Phi_1) +$$

$$\phi_T'\ \cos\Phi_1]$$

$$T23 = \sin(\Theta_T)[\sin(\phi_A)\ \sin(\Theta_A)\ \cos(\Delta\psi) - \cos(\phi_A)\ \sin(\Delta\psi)][\cos(\Phi_1) -$$

$$\phi_T'\ \sin(\Phi_1)] - [\sin(\phi_A)\ \sin(\Theta_A)\ \sin(\Delta\psi) + \cos(\phi_A)\ \cos(\Delta\psi)]\ .$$

$$[\sin(\Phi_1) + \phi_T'\ \cos(\Phi_1)] + \sin(\Phi_A)\ \cos(\Theta_A)\ \cos(\Theta_T)[\cos(\Phi_1) -$$

$$\phi_T'\ \sin(\Phi_1)]$$

$$T31 = \cos(\Theta_T)[\cos(\phi_A)\ \sin(\Theta_A)\ \cos(\Delta\psi) + \sin(\phi_A)\ \sin(\Delta\psi)] - \cos(\phi_A)\ \cos(\Theta_A)\ \sin(\Theta_T)$$

$$T32 = \sin(\Theta_T)[\cos(\phi_A)\ \sin(\Theta_A)\ \cos(\Delta\psi) + \sin(\phi_A)\ \sin(\Delta\psi)][\sin(\Phi_1) + \phi_T'\ \cos(\Phi_1)] + [\sin(\phi_A)\ \sin(\Theta_A)\ \sin(\Delta\psi) + \cos(\phi_A)\ \cos(\Delta\psi)]\ .\ [\cos(\Phi_1) - \phi_T'\ \sin(\Phi_1)] + \cos(\phi_A)\ \cos(\Theta_A)\ \cos(\Theta_T)[\sin(\Phi_1) + \phi_T'\ \cos(\Phi_1)]$$

$$T33 = \sin(\Theta_T)[\cos(\phi_A)\ \sin(\Theta_A)\ \cos(\Delta\psi) + \sin(\phi_A)\ \sin(\Delta\psi)][\cos(\Phi_1) - \phi_T'\ \sin(\Phi_1)] - [\cos(\phi_A)\ \sin(\Theta_A)\ \sin(\Delta\psi) - \sin(\phi_A)\ \cos(\Delta\psi)]\ .\ [\sin(\Phi_1) + \phi_T'\ \cos(\Phi_1)] + \cos(\phi_A)\ \cos(\Theta_A)\ \cos(\Theta_T)[\cos(\Phi_1) - \phi_T'\sin(\Phi_1)]$$

The primary use of the transformation is to transform $\dot{v}_{T_T}$ and $\dot{w}_{T_T}$, the perturbation velocities of the target, in the target's coordinate system, into $\dot{v}_{T_T}'$ and $\dot{w}_{T_T}'$, the same quantities, but in the attacker's coordinates. The equations for $\dot{v}_{T_T}'$ and $\dot{w}_{T_T}'$ are given by

$$\dot{v}_{T_T}' = [g\ \cos(\Phi_1)]\ \phi_T'\ \bar{j}_T = 8.05\ \phi_T' \qquad \text{(B1-4)}$$

$$\dot{w}_{T_T}' = [-g\ \sin\ (\Phi_1)]\ \phi_T'\ \bar{k} = -31.18\ \phi_T' \qquad \text{(B1-5)}$$

The transformation is found by using

$$\begin{pmatrix} 0 & \bar{i}_A \\ \dot{v}_{T_T}' & \bar{j}_A \\ \dot{w}_{T_T}' & \bar{k}_A \end{pmatrix} = \begin{bmatrix} T11 & T12 & T13 \\ T21 & T22 & T23 \\ T31 & T32 & T33 \end{bmatrix} \begin{pmatrix} 0 & \bar{i}_T \\ 8.05\ \phi_T' & \bar{j}_T \\ -31.18\phi_T' & \bar{k}_T \end{pmatrix} \text{(B1-6)}$$

and resulting in

$$\dot{v}_{T_T}' = (8.05\phi_T') \, T22 - (31.18\phi_T') \, T23 \qquad (B1\text{-}7)$$

$$\dot{w}_{T_T}' = (8.05\phi_T') \, T32 - (31.18\phi_T') \, T33 \qquad (B1\text{-}8)$$

APPENDIX E

Graphics Subroutines and Entities

The FORTRAN subroutines used to generate the simulation display, with a brief explanation of each, are presented below. A complete description, plus many other grahpics instructions, can be found in Reference 4.

1. GSPIN

GSPIN(ICA, NWDS, IER) establishes an array, called ICA, in a permanent core memory location. The dimension of ICA is set by NWDS, and ICA will provide the area for creating, displaying and manipulating the various entities. IER is an error return variable.

2. GDRAW

GDRAW(LX,LY, NPTS, LCODE) creates the code required to draw one or more straight lines between a set of points contained in the arrays LX and LY. NPTS determines the number of points and LCODE is a four digit integer that determines the exact display. For example, LCODE = 4000 tells the vector generator to draw line segments between the points, and the values in the Lx and LY arrays correspond to raster units on the display screen.

3. GVECT

GVECT(IVECT, X, Y) creates and updates the vector entity numbered IVECT. Lines are drawn from the point (X(1), Y(1)) to (X(2), Y(2)) to

(X(3),Y(3)) and so on for all the points contained in the storage arrays, X and Y.

4. GSTOP

As might be expected, this subroutine shuts off the simulation display to prevent screen burnout.

The graphics entities used in the programming are

ENTITY NUMBER	CONTENTS
1	Sight Symbol
2	Target Symbol
3	Upper Horizon Line
4	Middle Horizon Line
5	Lower Horizon Line

APPENDIX F

Simulation Computer Program

This appendix contains the FORTRAN computer program, called DISPLY, that simulates the multiaxis tracking task. First, the computer variables will be listed and defined, then the subroutines calleo in the program will be given with a brief description of each, and finally, a computer listing of DISPLY will be given.

F.1.1 Computer Variables

SYMBOL	MEANING
A, AAUG	Unaugmented and augmented system matrices, A and A_{aug}
AA	Array of A_i, used to calculate $\delta_A = \sum_{i=1}^{13} A_i \sin(\omega_i t+\phi_i)$
ADT	The matrix, A, multiplied by Δt
AGN	Aileron gain factor
AK	Weighting factor in Runga-Kutta integration routine
B, BAUG	Unaugmented and augmented system matrices, B and B_{aug}
BKx, BKu, Bx, Bu	The B matrix multiplied by the Kx and Ku matrices, and the x and u vectors, respectively.
C	The system matrix, C.
CK	Weighting factor in Runga-Kutta integration routine
CPYFLG	Logical variable used to decide if data will be taken
CX	The C matrix multipled by the X-vector
D	The system matrix, D
DAL	The target aileron deflection, in radians

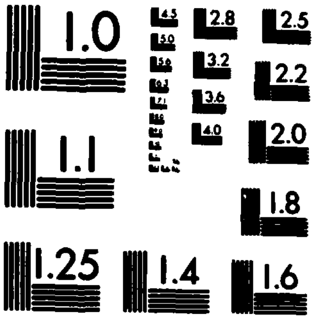
DT	The iteration step size, Δt
DTT	Array of the step sizes used in the Runga-Kutta integration routine
DX, DU	The D matrix multiplied by the X and Y vectors, respectively
E1, E2	Storage matrices used to calculate state transition matrices
EAT, EATX	$e^{A\Delta t}$ and $e^{A\Delta t}$ times the X vector, respectively
ERRT	Array of errors, used in Runga-Kutta routine
ETAU	$\int_0^{\Delta t} e^{A\tau} d\tau$
ETAUB, ETAUBU	ETAU multiplied by the B and BU matrices, respectively
HX,HX1,HX2,HX3	Storage arrays for horizon line endpoints (x-direction)
HY,HY1,HY2,HY3	Storage arrays for horizon line endpoints (y-direction)
I1, I2	Logical variables used to determine if printing should be done
IAUG	Logical variable used to decide if augmentation used
IMAGE	Storage array for graphics instructions
KU, KX	The augmentation matrices, K_u and K_x, respectively
KUU	The KU matrix times the U-vector
KXU, KXX	The KX matrix times U and X-vectors, respectively
OMEGA	Array of ω_i, used in $\delta_A = \sum_{i=1}^{13} A_i \sin(\omega_i t + \phi_i)$
PHIREF,PHIREL	ϕ_A, ϕ_T, the attacker's and target bank angles
POSX,POSX2,POSX3	Arrays for storage of vector endpoints (x-direction)
POSY,POSY2,POSY3	Arrays for storage of vector endpoints (y-direction)
RAD 755	75.5°, converted to radians
RDCF	Conversion factor, radians to degrees

AD-A125 302 COMPUTER SIMULATION OF A MULTIAXIS AIR-TO-AIR TRACKING TASK USING THE OPT..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH W A YUCUIS DEC 82

2/2

UNCLASSIFIED AFIT/CI/NR-82-62T F/G 9/2 NL

END
DATE
FILMED
[illegible]
DTIC


1.0
4.5
5.0
5.6
6.3
7.1
8.0
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

REF,REFSX,REFSY	Arrays for reference positions on the display screen, in raster units
RVALS	Array of pilot control inputs, in volts
RX,RY	Array of vector endpoints of reference cross
SX,SY	Array of vector endpoints of sight symbol
TAUR	Target aircraft roll constant, τ_R
TGT	Array of target motion variables, δ_A, p_T, Θ_T, ϕ_T
TGTX,TGTY	Array of vector endpoints of target symbol, with bank included
TX,TY	Array of vector endpoints of target symbol, with no banking
U	Array of attacking pilot's control inputs, $\bar{u}_p$
UAUG	Array of equivalent augmented control inputs, $\bar{u}_a$
X	Array of state vector, $\bar{x}$
XBUF	Storage matrix for data
Y	Array of the observation vector, $\bar{y}$

F.2. Simulation Subroutines

a. DASIN - converts analog voltage readings of strain gages, used to calculate stick forces, into a digital voltage, by averaging four readings over a 1 msec time period

b. FREMAT - allows for free format inputting of variables

c. FSDIO - transfers a data storage matrix, XBUF, to a buffer location, and then allows the disk to have direct access to the data

d. GMPRD, GMADD, GMSUB - multiplies, adds, and subtracts two matrices, respectively

e. TSECS - reads out the real time

f. TSET - starts the real time clock

F.3. Computer Listing

```
C** DISPLY - GIVES SIGHT/TARGET DISPLAY           F
      DIMENSION IMAGE(1500),RVALS(4)
      LOGICAL A$RDC,CPYFLG
      REAL AX(12),BU(12),X(12),Y(10),AAUG(12,12)
      REAL BAUG(12,3),U(3),XBUF(20,7)
      REAL EAT(12,12),ETAU(12,12),ETAUB(12,3),E1(12,12),E2(12,12)
      REAL ADT(12,12),EATX(12),ETAUBU(12)
      REAL POSX3(6),POSY3(6)
      REAL A(12,12),KXX(3),KUU(3),UAUG(3)
      REAL B(12,3),KX(3,12),KU(3,3)
      REAL BKX(12,12),BKU(12,3),C(10,12),CX(10),D(10,3)
      REAL DU(10),TGT(5),DTT(5),ERRT(5)
      REAL RX(4),RY(4),TX(6),TY(6),REF(4)
      REAL SX(8),SY(8),POSX(8),POSY(8),POSX2(2),POSY2(2)
      REAL HX(2),HY(2),HX1(2),HX2(2),HX3(2),HY1(2),HY2(2),HY3(2)
      REAL AK(4),BK(4),CK(4)
      REAL AA(13),OMEGA(13),PHIT2(13)
      INTEGER SI,DO,STACK(10),LO,SHIFT,CASE
      DATA AA/4*0.001,3*0.05,6*0.1/
      DATA OMEGA/.1885,.31416,.50265,.87965,1.44513,2.31628,3.07876,
     C     4.20973,5.78053,8.23097,11.2469,15.77079,23.93894/
      DATA NCHNLS/X'F000'/
      DATA SI/1/,LO/5/,SHIFT/8192/,DO/50/
      DATA STACK/14,1,1,7*0/
      DATA RX/460.0,560.0,510.0,510.0/
      DATA RY/765.0,765.0,715.0,815.0/
      DATA TX/-80.0,80.0,-20.0,0.0,0.0,20.0/
      DATA TY/0.0,0.0,0.0,30.0,30.0,0.0/
      DATA SX/-20.,-5.,0.,5.,20.,20.,-20.,-20./
      DATA SY/20.,20.,30.,20.,20.,-20.,-20.,20./
      DATA HX/-400.0,400.0/
      DATA HY/0.0,0.0/
      DATA REF/810.0,510.0,210.0,765.0/
      DATA AK/.5,.292892319,1.70710678,.166666667/
      DATA BK/2.,1.,1.,2./
      CK(1)=AK(1)
      CK(2)=AK(2)
      CK(3)=AK(3)
      CK(4)=.5
      REFSX=REF(2)-300.0*0.031
      REFSY=REF(4)-300.0*.127
      REFTX=REFSX
      REFTY=REFSY
      PI=3.141592654
      N=0
      RNUM=0.0
 250  RNUM=RNUM+1.0
       N=N+1
       PHIT2(N)=RNUM*PI/6.5
      IF(N.LT.13) GO TO 250
C          DEFINE DATA FILE AND SET UP IMPUT DEVICE
      DEFINE FILE DO(5000,12,U,IREC)
      CALL A$INT(STACK,'SY')
      ISY=IPUNM('SY')+32
      CASE='A'+SHIFT
```

```
C          SET CONCURRENT DISK I/O PARAMETERS
      ISECT=0
      INDEX=0
      IWRITE=2
      IMMED=1
      IREAD=1
      IWAIT=0
C          ZEROING OUT THE MATRICES
      DO 10 I=1,12
       DO 11 J=1,12
        E1(I,J)=0.0
        ETAU(I,J)=0.0
        A(I,J)=0.0
        IF(J.LT.4) KX(J,I)=0.0
        IF(J.LT.4) U(J)=0.0
        IF(J.LT.4) BKU(I,J)=0.0
        BKX(I,J)=0.0
        IF(I.LT.11) CX(I)=0.0
        X(I)=0.0
        IF(I.LT.11) Y(I)=0.0
        IF(J.LT.4) B(I,J)=0.0
        IF(J.LT.11) C(J,I)=0.0
 11   CONTINUE
 10   CONTINUE
      DO 12 I=1,10
       DO 13 J=1,3
       IF(I.LT.4) KU(I,J)=0.0
       IF(I.LT.6) TGT(I)=0.0
       IF(I.LT.6) DTT(I)=0.0
       IF(I.LT.6) ERRT(I)=0.0
 13    D(I,J)=0.0
 12   CONTINUE
C          IMPUT LINEARIZED AIRCRAFT/SIGHT DYNAMICS
      READ(SI,1) A(1,9),A(1,12),A(2,6),A(2,7),A(2,11),A(2,12)
      READ(SI,1) A(3,6),A(3,7),A(3,9),A(3,11),A(4,3),A(4,4)
      READ(SI,1) A(4,5),A(4,8),A(4,9),A(4,11),A(4,12)
      READ(SI,1) A(5,2),A(5,3),A(5,4),A(5,5),A(5,8),A(5,10)
      READ(SI,1) A(5,11),A(5,12),A(6,6),A(6,7),A(6,9),A(6,12)
      READ(SI,1) A(7,6),A(7,11),A(8,7),A(8,8),A(8,9)
      READ(SI,1) A(9,7),A(9,8),A(9,9),A(9,11),A(9,12)
      READ(SI,1) A(10,7),A(10,10),A(10,11),A(10,12),A(11,9),A(11,10)
      READ(SI,1) A(11,11),A(11,12),A(12,9),A(12,10),A(12,11),A(12,12)
      READ(SI,1) B(8,1),B(9,1),B(10,2),B(10,3)
      READ(SI,1) B(11,2),B(11,3),B(12,2),B(12,3)
      READ(SI,1) C(1,1),C(1,3),C(1,4),C(1,6),C(1,8)
      READ(SI,1) C(2,3),C(2,4),C(2,5),C(2,6),C(2,7),C(2,8)
      READ(SI,1) C(2,9),C(2,11),C(2,12),C(3,1),C(3,2),C(3,3)
      READ(SI,1) C(3,5),C(3,6),C(3,7),C(3,8),C(3,10)
      READ(SI,1) C(4,2),C(4,3),C(4,4),C(4,5),C(4,6),C(4,7)
      READ(SI,1) C(4,8),C(4,9),C(4,10),C(4,11),C(4,12)
      READ(SI,1) C(5,1),C(5,3),C(5,4),C(5,6),C(5,8)
      READ(SI,1) C(6,6),C(6,7),C(6,8),C(6,9],C(6,11)
      READ(SI,1) C(7,1),C(7,2),C(7,3),C(7,5)
      READ(SI,1) C(7,6),C(7,7),C(7,8),C(7,10),C(8,6),C(8,7)
      READ(SI,1) C(8,8),C(8,9),C(8,10),C(8,11),C(8,12)
      READ(SI,1) C(9,9),C(9,10),C(9,11),C(9,12),C(10,7)
      READ(SI,1) D(2,1),D(4,1),D(4,2),D(4,3)
      READ(SI,1) D(6,1),D(8,1),D(8,2),D(8,3)
      READ(SI,1) KX(1,1),KX(1,2),KX(1,3),KX(1,4),KX(1,5),KX(1,6)
      READ(SI,1) KX(1,7),KX(1,8),KX(1,9),KX(1,10),KX(1,11),KX(1,12)
      READ(SI,1) KX(2,1),KX(2,2),KX(2,3),KX(2,4),KX(2,5),KX(2,6)
      READ(SI,1) KX(2,7),KX(2,8),KX(2,9),KX(2,10),KX(2,11),KX(2,12)
      READ(SI,1) KX(3,1),KX(3,2),KX(3,3),KX(3,4),KX(3,5),KX(3,6)
      READ(SI,1) KX(3,7),KX(3,8),KX(3,9),KX(3,10),KX(3,11),KX(3,12)
      READ(SI,1) KU(1,1),KU(1,2),KU(2,1),KU(2,2),KU(3,1),KU(3,2)
      READ(SI,1) EGN,AGN,RGN,AUG
      READ(SI,2) I1
 1    FORMAT(8F10.4)
 2    FORMAT(8I10)
C               SET UP AND INITIALIZE DISPLAY AREA
      CALL GSPIN(IMAGE,1500,IER)
      CALL GDRAW(RX,RY,4,4000)
 3    CALL FREMAT
C          READ IN IF AUGMENTATION IS REQUIRED
C            (SET IAUG=0.0 IF NO AUGMENTATION REQUIRED)
C          READ IN DT, THE STEP INTERVAL SIZE
      WRITE(ISY,4)
 4    FORMAT('***************************************************',/,
     C       '***        ENTER AUG OPTION,DT                  ***',/,
     C       '***    AUG OPTION=0 MEANS NO AUG                ***',/,
     C       '***    AUG OPTION=1 MEANS LVL 2 AUG             ***',/,
     C       '***    DT=STEP SIZE-SHOULD BE > 0.0887          ***',/,
     C       '***************************************************')
      READ(ISY,5) IAUG,DT
 5    FORMAT(V)
      RNUM2=1.0
C             IF IAUG=0 NO AUGMENTATION REQUIRED AND
C             AAUG,BAUG ARE SET EQUAL TO A,B MATRICIES
      IF(IAUG.NE.0) GO TO 14
      DO 14 I=1,12
      DO 15 J=1,12
        AAUG(I,J)=A(I,J)
        IF(J.LT.4) BAUG(I,J)=B(I,J)
 15   CONTINUE
 14   CONTINUE
C             IF IAUG N.E. 0, AUGMENTATION IS REQUIRED AND
C             AAUG AND BAUG ARE CALCULATED
      IF(IAUG.EQ.0) GO TO 16
      CALL GMPRD(B,KX,BKX,12,3,12)
      CALL GMSUB(A,BKX,AAUG,12,12)
      CALL GMPRD(B,KU,BKU,12,3,3)
      CALL GMSUB(B,BKU,BAUG,12,3)
 16   CONTINUE
C             CALCULATE THE STATE TRANSITION MATRIX
      DO 17 I=1,12
       DO 18 J=1,12
 18    IF(I.EQ.J) E1(I,J)=DT
 17   CONTINUE
      DO 19 I=1,12
       DO 20 J=1,12
 20    ADT(I,J)=AAUG(I,J)*DT
 19   CONTINUE
 300  RNUM2=RNUM2+1.0
      CALL GMPRD(E1,ADT,E2,12,12,12)
      DO 21 I=1,12
       DO 22 J=1,12
 22    E2(I,J)=E2(I,J)/RNUM2
 21   CONTINUE
      DO 23 I=1,12
       DO 24 J=1,12
 24    IF(E2(I,J)-E1(I,J).GT.000001) GO TO 25
 23   CONTINUE
      GO TO 30
 25   DO 26 I=1,12
       DO 27 J=1,12
 27    ETAU(I,J)=ETAU(I,J)+E1(I,J)
 26   CONTINUE
      DO 28 I=1,12
       DO 29 J=1,12
 29    E1(I,J)=E2(I,J)
 28   CONTINUE
      GO TO 300
 30   CONTINUE
      CALL GMPRD(AAUG,ETAU,EAT,12,12,12)
      DO 31 I=1,12
       DO 32 J=1,12
 32    IF(I.EQ.J) EAT(I,J)=EAT(I,J)+1.0
 31   CONTINUE
      CALL GMPRD(ETAU,BAUG,ETAUB,12,12,3)
C               PRINTOUT OF A,B,C,D MATRICES
C               IF I1=0 MATRICES WILL NOT BE PRINTED
      IF(I1.EQ.0) GO TO 309
      WRITE(5,104)
      DO 33 I=1,12
       WRITE(5,102) A(I,1),A(I,2),A(I,3),A(I,4),A(I,5),A(I,6),A(I,7),
     C              A(I,8),A(I,9),A(I,10),A(I,11),A(I,12)
 33   CONTINUE
C                PRINTS B MATRIX
      WRITE(5,105)
      DO 34 I=1,12
 34    WRITE(5,103) B(I,1),B(I,2),B(I,3)
C               PRINTS C MATRIX
      WRITE(5,106)
      DO 35 I=1,10
       WRITE(5,102) C(I,1),C(I,2),C(I,3),C(I,4),C(I,5),C(I,6)
     C          ,C(I,7),C(I,8),C(I,9),C(I,10),C(I,11),C(I,12)
 35   CONTINUE
C              PRINTS D MATRIX
      WRITE(5,107)
      DO 36 I=1,10
 36    WRITE(5,103) D(I,1),D(I,2),D(I,3)
C             PRINTOUT KX,KU MATRICES
      WRITE(5,108)
      DO 37 I=1,12
 37    WRITE(5,103) KX(1,I),KX(2,I),KX(3,I)
      WRITE(5,109)
      DO 38 I=1,3
 38    WRITE(5,103) KU(I,1),KU(I,2),KU(I,3)
C             PRINTOUT AAUG AND BAUG MATRICES
      WRITE(5,110)
      DO 39 I=1,12
      WRITE(5,102) AAUG(I,1),AAUG(I,2),AAUG(I,3),AAUG(I,4),
     C             AAUG(I,5),AAUG(I,6),AAUG(I,7),AAUG(I,8),AAUG(I,9),
     C             AAUG(I,10),AAUG(I,11),AAUG(I,12)
 39   CONTINUE
      WRITE(5,111)
      DO 40 I=1,12
 40    WRITE(5,103) BAUG(I,1),BAUG(I,2),BAUG(I,3)
C                PRINTOUT OF TRANSITION MATRICIES
C     WRITE(5,112) DT,IAUG,RNUM2
      WRITE(5,113)
      DO 41 I=1,12
 41    WRITE(5,114) (ETAU(I,L),L=1,12)
      WRITE(5,115)
      DO 42 I=1,12
 42    WRITE(5,116) (ETAUB(I,L),L=1,3)
      WRITE(5,117)
      DO 43 I=1,12
 43    WRITE(5,114) (EAT(I,L),L=1,12)
 309  CONTINUE
C                SET INITIAL VALUES
      RDCF=57.29578
      RAD755=-75.5/RDCF
      CPYFLG=.FALSE.
      TAUR=2.0
      N2=-1
      N=0
      VTDT=0.0
      WTDT=0.0
C     X(1)=.412
      X(2)=78.0
      X(3)=-300.0
 330  CONTINUE
      T=0.0
      DO 500 I=1,4
       TGT(I)=0.0
       DTT(I)=0.0
 500  ERRT(I)=0.0
C              START THE CLOCK AND READ STICK AND RUDDER VOLTAGES
C                    AND CONVERT TO DEFLECTION VALUES
      CALL TSET(T)
 301  CALL DASIN(NCHNLS,RVALS)
      CALL TSECS(RT1)
C          CHECK FOR IMPUT FROM KEYBOARD
      IF(.NOT.(A$RDC(STACK,LTR))) GO TO 302
      IF(LTR.GE.CASE) LTR=LTR-SHIFT
      IF(LTR.EQ.'C') GO TO 304
      IF(LTR.EQ.'S') GO TO 1000
```

```
  302 CONTINUE
      U(1)=RVALS(1)*EGN
      U(2)=-RVALS(2)*AGN
      U(3)=RVALS(3)*RGN
C               CALCULATE X VECTOR
      DO 44 I=1,12
       EATX(I)=0.0
   44 ETAUBU(I)=0.0
      DO 45 I=1,12
       DO 46 J=1,12
   46  EATX(I)=EAT(I,J)*X(J)+EATX(I)
      ETAUBU(I)=ETAUBU(I)+ETAUB(I,1)*U(1)+ETAUB(I,2)*U(2)+
     C          ETAUB(I,3)*U(3)
   45 X(I)=ETAUBU(I)+EATX(I)
      PHIREF=X(7)
C             CALCULATE THE Y VECTOR
      DO 47 I=1,10
       CX(I)=0.0
   47 DU(I)=0.0
      DO 48 I=1,10
       DO 49 J=1,12
   49  CX(I)=CX(I)+C(I,J)*X(J)
   48 CONTINUE
      DO 50 I=1,10
       DU(I)=D(I,1)*U(1)+D(I,2)*U(2)+D(I,3)*U(3)+DU(I)
   50 Y(I)=DU(I)+CX(I)
C                 CALCULATE TARGET MOTION
      DAL=0.0
      DO 92 I=1,13
   92  DAL=DAL+AA(I)*SIN(OMEGA(I)*T+PHIT2(I))
      DT2=DT/2.
      DTT(2)=-TGT(2)/TAUR+DAL
      DTT(3)=-.161*TGT(4)
      DTT(4)=TGT(2)+.161*TGT(3)
      J=1
  405 CONTINUE
      DO 91 I=2,4
       R11=DT*DTT(I)
       R22=AK(J)*(R11-BK(J)*ERRT(I))
       TGT(I)=TGT(I)+R22
       R22=R22*3.0
   91 ERRT(I)=ERRT(I)+R22-CK(J)*R11
      IF(J.EQ.4) GO TO 406
      J=J+1
      IF(J.NE.3) T=T+DT2
      DTT(2)=-TGT(2)/TAUR+DAL
      DTT(3)=-.161*TGT(4)
      DTT(4)=TGT(2)+.161*TGT(3)
      GO TO 405
  406 CONTINUE
      PHIREL=TGT(4)
C                 CALCULATES TRANSFORMATION MATRIX
      C1=COS(X(1))
      C2=COS(X(6))
      C3=COS(X(7))
      C4=COS(TGT(3))
      C5=SIN(X(1))
      C6=SIN(X(6))
      C7=SIN(X(7))
      C8=SIN(TGT(3))
      C9=.9682+.25*TGT(4)
      C10=.9682-.25*TGT(4)
      C11=.25-.9682*TGT(4)
      C12=C2*C1*C8
      C13=C1*C6*C7*C8
      C14=C5*C6*C7
      C15=C2*C4*C7
      C16=C1*C3*C6
      C17=C2*C3*C4
      T11=C1*C2*C4+C6*C8
      T12=C12*C9*C2*C5*C11-C4*C6*C9
      T13=C12*C11*C2*C5*C10-C4*C6*C11
      T21=C4*C7*C6*C1-C3*C5-C2*C7*C8
      T22=C13-C3*C5*C9+C14+C1*C3*C11+C9*C15
      T23=C13-C3*C5*C11-C14+C1*C3*C9+C11*C15
      T31=C4*C16+C5*C7-C3*C6*C8
      T32=C8*C16+C5*C7*C9+C14+C1*C3*C11+C9*C17
      T33=C8*C16+C5*C7*C11-C3*C5*C6-C1*C7*C9+C11*C17
      VTDT=8.05*T22*TGT(4)-31.18*T23*TGT(4)
      WTDT=8.05*T32*TGT(4)-31.18*T33*TGT(4)
      X(2)=X(2)+VTDT
      X(3)=X(3)+WTDT
C              SIGHT POSITION UPDATED
      C24=300.0*Y(5)
      C25=300.0*Y(7)
      DO 52 I=1,8
       POSX(I)=SX(I)+REFSX+C25
       POSY(I)=SY(I)+REFSY-C24
 52   CONTINUE
      CALL GVECT(1,POSX,POSY,DUMY,8,0000)
C              TARGET POSITION UPDATED
      CPT=COS(PHIREL)
      SPT=SIN(PHIREL)
      C22=300.0*X(4)
      C23=300.0*X(5)
      TGTX=REFTX+C23
      TGTY=REFTY-C22
      DO 54 I=1,6
       POSX3(I)=TX(I)*CPT+TY(I)*SPT+TGTX
       POSY3(I)=-TX(I)*SPT+TY(I)*CPT+TGTY
 54   CONTINUE
      CALL GVECT(2,POSX3,POSY3,DUMY,6,4000)
C              HORIZON LINES POSITIONS UPDATED
      PHI=RAD755-PHIREF
      C20=300.0*SIN(PHI)
      C21=300.0*COS(PHI)
      DO 53 I=1,2
       POSX2(I)=HX(I)*COS(PHI)+HY(I)*SIN(PHI)+510.0
       POSY2(I)=-HX(I)*SIN(PHI)+HY(I)*COS(PHI)
       HX1(I)=POSX2(I)+C20
       HX2(I)=POSX2(I)
       HX3(I)=POSX2(I)-C20
       HY1(I)=POSY2(I)+REF(2)+C21
       HY2(I)=POSY2(I)+REF(2)
       HY3(I)=POSY2(I)+REF(2)-C21
  53  CONTINUE
      CALL GVECT(3,HX1,HY1,DUMY,2,0000)
      CALL GVECT(4,HX2,HY2,DUMY,2,0000)
      CALL GVECT(5,HX3,HY3,DUMY,2,0000)
C             COPY PARAMETERS TO DISK(EVERY OTHER VALUE)
      IF(ICOPY.NE.1) GO TO 307
      N1=-N2
      IF(N1.GT.0) GO TO 308
      INDEX=INDEX+1
      XBUF(1,INDEX)=T
      XBUF(2,INDEX)=X(1)*RDCF
      XBUF(3,INDEX)=X(2)
      XBUF(4,INDEX)=X(3)
      XBUF(5,INDEX)=X(4)*RDCF
      XBUF(6,INDEX)=X(5)*RDCF
      XBUF(7,INDEX)=X(6)*RDCF
      XBUF(8,INDEX)=X(7)*RDCF
      XBUF(9,INDEX)=X(8)*RDCF
      XBUF(10,INDEX)=X(9)*RDCF
      XBUF(11,INDEX)=X(10)*RDCF
      XBUF(12,INDEX)=X(11)*RDCF
      XBUF(13,INDEX)=X(12)*RDCF
      XBUF(14,INDEX)=Y(1)*RDCF
      XBUF(15,INDEX)=Y(2)*RDCF
      XBUF(16,INDEX)=Y(3)*RDCF
      XBUF(17,INDEX)=Y(4)*RDCF
      XBUF(18,INDEX)=Y(5)*RDCF
      XBUF(19,INDEX)=Y(6)*RDCF
      XBUF(20,INDEX)=Y(7)*RDCF
      XBUF(21,INDEX)=Y(8)*RDCF
      XBUF(22,INDEX)=Y(9)
      XBUF(23,INDEX)=U(1)
      XBUF(24,INDEX)=U(2)
      XBUF(25,INDEX)=U(3)
      XBUF(26,INDEX)=DAL
      XBUF(27,INDEX)=TGT(4)*RDCF
      XBUF(28,INDEX)=TGT(2)
      IF(INDEX.LT.7) GO TO 308
      CALL FSDIO(IWRITE,IMMED,ISECT,XBUF,IRC)
      IF(IRC.NE.0) GO TO 1000
      ISECT=ISECT+1
      INDEX=0
 308  CONTINUE
      N2=N1
 307  CONTINUE
      CALL TSECS(RT7)
C       TELL COMPUTER TO WAIT TILL REAL TIME
      TC=RT7-RT1
      IF(TC.GT.DT) WRITE(5,119) T,TC,DT
      CALL TWAIT(T)
      GO TO 301
C********************************************
C*********   END OF ITERATION LOOP   ********
C********************************************
C            KEYBOARD IMPUT DIECTIVES
C        IF KEYBOARD IMPUTS 'C', DATA IS WRITTEN TO DISK
 304  ICOPY=1
      CPYFLG=.TRUE.
      WRITE(5,110)
      GO TO 330
C             STOP SIMULATION AND PRESENT DESIRED OUTPUT
 1000 CONTINUE
      CALL GSTOP
C            MARK END OF DISK FILE BY SETTING T=-1.0
      T=-1.0
      IF(.NOT.CPYFLG) GO TO 2000
      INDEX=INDEX+1
      XBUF(1,INDEX)=T
      CALL FSDIO(IWRITE,IWAIT,ISECT,XBUF,IRC)
       CALL F$OPN(2)
      ISECT=ISECT+1
      DO 57 ISC=1,ISECT
       ISEC=ISC-1
       CALL FSDIO(IREAD,IWAIT,ISEC,XBUF,IRC)
       IF(IRC.NE.0) GO TO 2000
       DO 58 I=1,7
      IF(IAUG.EQ.0) GO TO 320
      KXX(1)=0.0
      KXX(2)=0.0
      KXX(3)=0.0
      KUU(1)=0.0
      KUU(2)=0.0
      KUU(3)=0.0
 320  CONTINUE
        WRITE(2,100) (XBUF(L,I),L=1,5)
        WRITE(2,100) (XBUF(L,I),L=6,10)
        WRITE(2,100) (XBUF(L,I),L=11,15)
        WRITE(2,100) (XBUF(L,I),L=16,20)
        WRITE(2,100) (XBUF(L,I),L=21,25)
C              CALCULATES AUGMENTED STICK IMPUT,IF IAUG N.E. 0
      UAUG(1)=0.0
      UAUG(2)=0.0
      UAUG(3)=0.0
      IF(IAUG.EQ.0) GO TO 322
      DO 59 J=1,3
       DO 60 K=5,13
 60     KXX(J)=KXX(J)+KX(J,K-1)*XBUF(K,I)/RDCF
       KXX(J)=KXX(J)+KX(J,1)*XBUF(2,I)/RDCF+KX(J,2)*XBUF(3,I)+
     C       KX(J,3)*XBUF(4,I)
       KUU(J)=KUU(J)+KU(J,1)*XBUF(23,I)+KU(J,2)*XBUF(24,I)+
     C       KU(J,3)*XBUF(25,I)
 59   UAUG(J)=KXX(J)+KUU(J)
 322  CONTINUE
        WRITE(2,100) XBUF(26,I),XBUF(27,I),UAUG(1),UAUG(2),UAUG(3)
        IF (XBUF(1,I) .LT. 0.0) GO TO 1099
 58   CONTINUE
 57   CONTINUE
 1099 WRITE(2,101)
 2000 CONTINUE
C                THE WRITE STATEMENT FORMATS
 100  FORMAT(5X,5E15.7)
 101  FORMAT(' END ',75X)
 102  FORMAT(12F9.4)
 103  FORMAT(3F12.7)
 104  FORMAT(////45X,'THE A MATRIX'/)
 105  FORMAT(//15X,'THE B MATRIX'/)
 106  FORMAT(//////////45X,'THE C MATRIX'/)
 107  FORMAT(//15X,'THE D MATRIX'/)
 108  FORMAT(//////////10X,'THE KX AUGMENTATION MATRIX'/)
 109  FORMAT(//10X,'THE KU AUGMENTATION MATRIX'/)
 110  FORMAT(//////////45X,'THE AAUG MATRIX'/)
 111  FORMAT(//15X,'THE BAUG MATRIX'/)
C112  FORMAT(/////,'THE VALUE OF DT= ',F5.2,/,'THE VALUE OF IAUG= ',
C    C    I5,/,'THE TOTAL NUMBER OF ITERATIONS WAS= ',F7.1,//)
 113  FORMAT(/////45X,'THE EXP TAU MATRIX'/)
 114  FORMAT(12F9.4)
 115  FORMAT(//5X,'THE EXP TAU TIMES B MATRIX'/)
 116  FORMAT(3(1X,F9.4))
 117  FORMAT(//////////45X,'THE TRANSITION MATRIX'/)
 118  FORMAT('***** DATA WILL BE COPIED TO DISK NOW *****')
 119  FORMAT('AT TIME= ',F8.4,' SECONDS, THE ITERATION CYCLE TIME'
     C        'WAS= ',F8.4,' ,WHICH EXCEEDS THE SET ITERATION TIME'
     C        'OF ',F8.4,' SECONDS',/)
      STOP
      END
```

APPENDIX G

Plotting Program

This appendix contains information about the computer routine, WAYPLT, which calculates RMS statistics and plots. Variables used in the program will first be defined and then the subroutines called will be briefly explained. Finally, a computer listing of WAYPLT is given.

G.1 Computer Variables - small x represents any variable

SYMBOL	DEFINITION
AVDESQ	Sum of the variable, squared, and divided by the total number of points, $(\sum_1^N x^2)/N$
AVG	Average of the variable, $(\sum_1^N x)/N$
DEF1, DEF2	Last value and present value of control input, $\bar{u}_p$
DESM	Sum of the variable, $\sum_1^N x$
DESQ	Sum of the variable, squared, $\sum_1^N x^2$
IDONE	Logical variable to decide if another variable should be selected
IPLT	Logical variable to decide if variables are plotted
IRATE	Logical variable to decide if control inputs are calculated
IRMS	Logical variable to decide if RMS statistics are calculated

PTS Total number of points, N

RMDE RMS value of the variable, $((\sum_1^N x^2)/N)^{1/2}$

SDDE Standard deviation of the variable, $((\sum_1^N x)/N)^2|)^{1/2}$

SQAVDE Sum of the variable, divided by the total number of points, and then squared, $((\sum_1^N x)/N)^2$

T1, T2 Last value and present value of time, respectively

VNAME Storage array for variable names used on plots

X* Storage array for plotting points in the x-direction

XBUF Storage array of data

Y* Storage array for plotting points in the y-direction

*the integer number assigned to each parameter can be found in Appendix H.

G.2. Subroutines

AXES(XSPEC,XTITLE, YSPEC, YTITLE) - draws and labels plot axes based on scaling factors calculated in S/R SCAN

XSPEC, YSPEC - real number in form NN.D

if NN 0, NN = number of characters in axis title
if NN 0, no axis drawn
D = number of decimal places for numbers annotated on axes

XTITLE, YTITLE - array containing characters of axes titles

DRAW(X, Y, N, IDS) - draws a line between a sequential set of coordinates, or draws a point at each coordinate

X,Y - real or integer arrays of coordinates
N - integer value, whose absolute value equals the number of coordinates if N = 0, plotting is terminated

IDS - integer data specification, in the form txyc
    t = line type
        t = 0, draws straight lines between points
        t = 9, draws straight line to the last point
    x,y - type of data in X,Y arrays
        0 = real value that does not have to be scaled
        4 = real value that must be scaled
    C-line width
        0 = previous width

F$OPN(1) - opens up the assigned storage location so the information can be manipulated by FORTRAN programming, such as READ and WRITE commands

FREMAT - allows usage of FORTRAN free-format READ and WRITE statements

LINK - terminates plotting program and allows computer system to call VPLOT program, which generates the plots

SCAN(X, Y, N, IDS) - scans the data in the storage arrays and calculates scaling factors to map coordinates onto the electrostatic printer/plotter

X, Y - real or integer arrays of coordinates
N - integer value, whose absolute value equals the number of
    coordinates if N < 0, determines minimum and maximum values
    of coordinates and calculates scale factors
IDS-integer data specification, in the form xy0
    x,y - type of data in X,Y arrays
    4 = real value that must be scaled

G.3 Computer Listing

```
C** WAYPLT - PLOTS DISPLY DATA VS TIME          F                  11/24/82
      DIMENSION X(900),Y(900),XBUF(30),VNAME(33)
      DOUBLE PRECISION DESM,DESQ,AVDESQ,SQAVDE,SDDE,RMDE,AVG
      DATA VNAME/'TIME','PSI ',' VT ',' WT ','T EL',
     C           'T AZ','TETA ','PHI ','ALPH',' Q  ',
     C           'BETA',' P  ',' R  ','E EL','E RT',
     C           'E AZ','EA R','S EL','SE R','S AZ',
     C           'SA R','TACC',' DE ',' DA ',' DR ',
     C           'DAL ','PHIT','DE A','DA A','DR A',
     C           'DE R','DA R','DR R'/
 5    WRITE(7,10)
 10   FORMAT('*****************************************************',/,
     C       '***** ENTER IX,IY,IPLT,IRMS,IRATE                *****',/,
     C       '*****  IX,IY=SUBSCRIPT OF PARA BEING PLOTTED     *****',/,
     C       '*****  IPLT=0 IF NO PLOTS,  IPLT=1 IF PLOT REQ   *****',/,
     C       '*****  IRMS=0 IF RMS NOT CALC, IRMS=1 IF CALC    *****',/,
     C       '*****  IRATE=1 IF STICK RATES CALC,ELSE IRATE=0 *****',/,
     C       '*****************************************************')
C$$$$$$$$$$  VALUES OF IY: 1=T, 2=PSI, 3=VT, 4=WT, 5=TGT EL    $$$$$$$
C$$$$$$$$$$    6=AZ T, 7=THETA, 8=PHI, 9=ALPHA, 10=Q, 11=BETA $$$$$$$
C$$$$$$$$$$    12=P, 13=R, 14=EL ERR, 15=EL ERR RT, 16=E AZ   $$$$$$$
C$$$$$$$$$$    17=AZ ERR RT, 18=SGT EL ERR, 19=SGT E RT       $$$$$$$
C$$$$$$$$$$    20=S AZ ERR, 21=SAZ E RT, 22=TGT ACC, 23=DE    $$$$$$$
C$$$$$$$$$$    24=DA, 25=DR, 26=DA TGT ,27=PHI TGT            $$$$$$$
C$$$$$$$$$$    28=DE AUG, 29=DA AUG, 30=DR AUG                $$$$$$$
C$$$$$$$$$$    31=DE RATE, 32=DA RATE, 33=DR RATE             $$$$$$$
      CALL FREMAT
      READ(7,11) IX,IY,IPLT,IRMS,IRATE
 11   FORMAT(V)
      ICNT=0
      IY2=IY
      IF(IRATE.EQ.1) IY2=IY+8
      T1=0.0
      DEF1=0.0
      DESM=0.0
      DESQ=0.0
      CALL F$OPN(1)
 12   READ(1,20) (XBUF(L),L=1,5)
      READ(1,20) (XBUF(L),L=6,10)
      READ(1,20) (XBUF(L),L=11,15)
      READ(1,20) (XBUF(L),L=16,20)
      READ(1,20) (XBUF(L),L=21,25)
      READ(1,20) (XBUF(L),L=26,30)
      IF(XBUF(1).EQ.-1) GO TO 100
      ICNT=ICNT+1
      IF(IRATE.EQ.0) GO TO 15
       X(ICNT)=XBUF(IX)
       T2=XBUF(IX)
       DT=T2-T1
       T1=T2
       DEF2=XBUF(IY)
       DELTA=DEF2-DEF1
       DEF1=DEF2
       Y(ICNT)=DELTA/DT
       GO TO 16
```

```
  15  CONTINUE
      X(ICNT)=XBUF(IX)
      Y(ICNT)=XBUF(IY)
  16  CONTINUE
      IF(IRMS.EQ.0) GO TO 30
       DESM=DESM+Y(ICNT)
       DESQ=DESQ+Y(ICNT)*Y(ICNT)
  30  CONTINUE
  20  FORMAT(5X,5E15.7)
      GO TO 12
C           CALCULATE RMS AND STD DEV, IF REQ
 100  IF(IRMS.EQ.0) GO TO 40
      PTS=FLOAT(ICNT)
      AVG=DESM/PTS
      AVDESQ=DESQ/PTS
      SQAVDE=(DESM/PTS)**2.0
      SDDE=SQRT(ABS(AVDESQ-SQAVDE))
      RMDE=SQRT(AVDESQ)
      WRITE(5,21) VNAME(IY2),AVG,VNAME(IY2),RMDE,VNAME(IY2),SDDE,ICNT
  21  FORMAT('  THE AVG VALUE OF ',A4,' IS=',F11.5,/,
     C       '  THE RMS VALUE OF ',A4,' IS=',F11.5,/,
     C       '  THE STD DEV OF   ',A4,' IS=',F11.5,/
     C       '  THE NUM OF POINTS=',I5,/)
  40  CONTINUE
C           START PLOTTING
      IF(IPLT.EQ.0) GO TO 41
      CALL SCAN(X,Y,-ICNT,0440)
      CALL AXES(4.4,VNAME(IX),4.4,VNAME(IY2))
      CALL DRAW(X,Y,ICNT,0440)
      CALL DRAW(0.0,0.0,1,9000)
  41  CONTINUE
      WRITE(7,13)
  13  FORMAT('***************************************************',/,
     C       '*****      ENTER IDONE                        *****',/,
     C       '***** IF IDONE=0,START PLOTTING               *****',/,
     C       '***** IF IDONE=1,SELECT NEW PARAMETER TO PLOT *****',/,
     C       '***************************************************',/)
      READ(7,14) IDONE
  14  FORMAT(V)
      IF(IDONE.EQ.1) GO TO 5
      CALL LINK('VPLOT ')
      STOP
      END
```

APPENDIX H

Simulation Results

Appendix H contains the identification procedure used to store the simulation data and defines the variables stored. In addition, the complete set or RMS statistics used to analyze the simulation are tabulated. Finally, the comments of each individual who flew the simulation are given.

H1. Datafile Identification Procedure

The four individuals who took part in the simulation are identified by:

(1) DJB - Lt. Colonel in the US Air Force, with 4000 hours of flying time in helicopters, transports, and as a test pilot.

(2) DAW - Captain in the US Air Force, with 1200 hours of KC-135 flying time.

(3) BJH - Purdue graduate student with 25 hours of civilian flying.

(4) WAY - Captain in the US Air Force, with 900 hours of F-4 flying time.

The data obtained in the simulation was stored on disk using the storage location identifier of XXXABC, where

XXX = individuals initials, given above
A = augmentation identifier
if A = 0, no augmentation
if A = 1, augmentation

B = set equal to zero for this simulation
C = the simulation run number for that particular augmentation configuration

For example, DJB101 contains data for individual DJB's first simulation run, with augmentation. The variables stored in the datafiles are given below

Storage Location	Analytic Identifier	Computer Variable Name	Meaning	Units
1	t	T	Time	sec
2	$\Delta\psi$	X(1)	Heading difference	deg
3	V_T	X(2)	Target velocity component	ft/sec
4	W_T	X(3)	Target velocity component	ft/sec
5	β_{EL}	X(4)	Target elevation angle	deg
6	β_{AZ}	X(5)	Target azimuth angle	deg
7	θ	X(6)	Attacker pitch angle	deg
8	ϕ	X(7)	Attacker bank angle	deg
9	α	X(8)	Attacker angle of attack	deg
10	q	X(9)	Stability axis angular rate	deg/sec
11	β	X(10)	Attacker sideslip angle	deg
12	p	X(11)	Stability axis angular rate	deg/sec
13	r	X(12)	Stability axis angular rate	deg/sec
14	ϵ_{EL}	Y(1)	Elevation error	deg
15	$\dot{\epsilon}_{EL}$	Y(2)	Elevation error rate	deg/sec
16	ϵ_{AZ}	Y(3)	Azimuth error	deg
17	$\dot{\epsilon}_{AZ}$	Y(4)	Azimuth error rate	deg/sec
18	λ_{EL}	Y(5)	Sight elevation angle	deg
19	$\dot{\lambda}_{EL}$	Y(6)	Sight elevation angle rate	deg/sec
20	λ_{AZ}	Y(7)	Sight azimuth angle	deg

Storage Location	Analytic Identifier	Computer Variable Name	Meaning	Units
21	$\dot{\lambda}_{AZ}$	Y(8)	Sight azimuth error rate	deg/sec
22	A_y	Y(9)	Attacker's lateral accel.	ft/sec
23	$\delta_{E_{st}}$	u(1)	Elevator deflection	in.
24	$\delta_{A_{st}}$	u(2)	Aileron deflection	in.
25	$\delta_{R_{ped}}$	u(3)	Rudder deflection	in.
26	δ_{A_T}	DAL	Target aileron deflection	radians
27	ϕ_T	TGT(4)	Target perturbation bank angle	deg
28	$\delta_{E_{aug}}$	$U_{aug}(1)$	Augmentation elevator deflection	in.
29	$\delta_{A_{aug}}$	$U_{aug}(2)$	Augmentation aileron deflection	in.
30	$\delta_{R_{aug}}$	$U_{aug}(3)$	Augmentation rudder deflection	in.

H2. RMS Statistics

This section contains tabulated results of RMS statistics calculated for each individual.

Table 18

DJB Simulation Results (RMS)

PARAMETER	DJB001	DJB002	DJB101	DJB102
ϵ_{EL}	.79	.59	.31	.33
ϵ_{AZ}	3.55	2.98	1.37	1.31
V_T	19.52	16.51	6.17	6.11
W_T	28.64	15.71	13.77	27.76
α	.79	.49	.29	.67
β	.32	.28	.35	.40
λ_{EL}	.84	.42	.30	.84
λ_{AZ}	.54	.54	.17	.20
Θ	8.71	10.96	10.96	9.11
ϕ	10.56	11.48	10.66	9.54
p	11 66	10.29	3.04	2.66
q	1.78	1.28	.57	1.10
r	1.04	1.18	.47	.44
δ_E	.22	.13	.10	.23
δ_A	.68	.58	.18	.18
δ_R	.36	.19	.36	.36
$\dot{\delta}_E$	.79	.57	.36	.41
$\dot{\delta}_A$	2.60	2.58	.55	.48
$\dot{\delta}_R$	1.87	1.01	.96	.92
A_y	3.95	2.69	2.36	1.61
ϕ_T(AUG/RMS)	-.39/5.28	-.40/5.23	-.40/5.23	-.40/5.21

Table 19

BJH Simulation Results (RMS)

PARAMETER	BJH001	BJH002	BJH101	BJH102
ϵ_{EL}	1.33	.76	.57	.82
ϵ_{AZ}	4.55	3.86	1.30	1.48
V_T	21.63	17.40	5.55	5.81
W_T	32.23	19.35	29.78	29.45
α	1.01	.54	.73	.80
β	.41	.36	.25	.27
λ_{EL}	.91	.49	.88	.88
λ_{AZ}	.58	.47	.15	.16
Θ	9.03	10.94	9.14	9.06
ϕ	10.63	11.60	9.93	9.85
p	9.52	8.10	2.24	2.41
q	2.54	1.52	1.21	1.38
r	1.43	1.30	.23	.25
δ_E	.28	.14	.25	.29
δ_A	.43	.39	.20	.20
δ_R	~0	~0	.23	.25
$\dot{\delta}_E$	1.13	.64	.45	.74
$\dot{\delta}_A$	1.79	1.63	.41	.52
$\dot{\delta}_R$	~0	~0	.32	.35
A_y	2.06	1.82	.97	.96
ϕ_T(AUG/RMS)	-.40/5.22	-.40/5.26	-.31/5.34	-.39/5.29

Table 20

DAW Simulation Results (RMS)

PARAMETER	DAW001	DAW002	DAW101	DAW102
ϵ_{EL}	.63	0.74	.55	.42
ϵ_{AZ}	3.19	3.81	1.22	1.47
V_T	12.71	17.02	5.68	6.55
W_T	23.07	15.76	23.78	20.31
α	.57	.53	.53	.39
β	.29	.32	.36	.30
λ_{EL}	.67	.38	.71	.58
λ_{AZ}	.37	.46	.15	.21
Θ	9.57	11.14	9.31	10.58
ϕ	10.37	11.20	10.07	10.56
p	5.64	7.58	2.54	3.11
q	1.11	1.15	.93	.79
r	.71	.94	.37	.39
δ_E	.14	.13	.19	.15
δ_A	.27	.36	.13	.20
δ_R	.09	.15	.35	.29
$\dot{\delta}_E$	.27	.36	.23	.23
$\dot{\delta}_A$	1.01	1.29	.19	.32
$\dot{\delta}_R$	.29	.35	.65	.72
A_y	1.09	1.57	1.38	1.17
ϕ_T(AUG/RMS)	-.39/5.28	-.34/5.17	-.38/5.33	-.39/5.26

Table 21

WAY Simulation Results (RMS)

PARAMETER	WAY001	WAY002	WAY101	WAY102
ϵ_{EL}	1.01	.98	.39	.51
ϵ_{AZ}	4.41	4.05	1.24	1.11
V_T	21.52	19.64	6.17	5.57
W_T	23.59	26.05	23.24	28.56
α	.80	.81	.49	.69
β	.71	.67	.45	.40
λ_{EL}	.60	.71	.69	.86
λ_{AZ}	.60	.57	.16	.15
θ	10.36	9.55	10.10	9.10
ϕ	11.85	10.80	10.00	9.60
p	13.82	13.57	2.65	2.03
q	2.79	2.52	.88	1.12
r	2.86	2.80	.30	.24
δ_E	.28	.27	.17	.25
δ_A	.74	.70	.13	.21
δ_R	~ 0	~ 0	.42	.37
$\dot{\delta}_E$	1.64	1.55	.20	.29
$\dot{\delta}_A$	3.41	3.44	.29	.31
$\dot{\delta}_R$	~ 0	~ 0	.47	.26
A_y	3.37	3.15	3.09	3.41
ϕ_T(AUG/RMS)	-.40/5.22	-.31/5.09	-.40/5.22	-.40/5.26

H.3 Pilot Comments

This section contains the comments of each subject.

Pilot DJB:

Was the augmentation system helpful? Yes

Any undesireable or annoying characteristics? No

How would you rate the task difficulty, from 1-10, where

1 = no inputs required

5 = reasonable difficulty for the task

10 = uncontrollable

Unaugmented system? 5-6

Augmented system? 2-3

Comments: Fairly realistic task

Pilot BJH:

Was the augmentation system helpful? Yes

Any undesireable or annoying characteristics? No

How would you rate the task difficulty, from 1-10, where

1 = no inputs required

5 = reasonable difficulty for the task

10 = uncontrollable

Unaugmented system? 6-7

Augmented system? 2-3

Comments: Fun simulation. Without augmentation, you had to anticipate the target motion by watching its bank.

Pilot DAW:

Was the augm-ntation system helpful? Yes

Any undesireable or annoying characteristics? No

How would you rate the task difficulty, from 1-10 where

1 = no inputs required

5 = reasonable difficulty for task

10 = uncontrollable

Unaugmented system? 7

Augmented system? 3

Comments: Initially, when using the rudder and aileron to track the target, the two control surfaces were actually fighting each other.

Pilot WAY:

Was the augmentation system helpful? Yes

Any undesireable or annoying characteristics? No

How would you rate the task difficulty, from 1-10, where

1 = no inputs required

5 = reasonable difficulty for the task

10 = uncontrollable

Unaugmented system? 7

Augmented system? 2

Comments: Difficult to maintain azimuth control without augmentation.

LME

8